ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΕΦΑΡΜΟΣΜΕΝΩΝ ΜΑΘΗΜΑΤΙΚΩΝ ΚΑΙ ΦΥΣΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

Ταξινόμηση αρχείων MIDI με χρήση Ετερογενών Γράφων και Νευρωνικών Δικτύων Γράφων

Εθνικό Μετσόβιο Πολυτεχνείο
AILS Lab

Βαμβακάς Παναγιώτης

Επιβλέποντες:
Συμβώνης Αντώνιος, ΣΕΜΦΕ
Στάμου Γεώργιος, ΗΜΜΥ

Αθήνα, 6 Ιουλίου 2023

.....

Βαμβακάς Παναγιώτης
vamvpanos@gmail.com

Περίληψη

Η παρούσα διατριβή διερευνά την εφαρμογή των Ετερογενών Γράφων και των Νευρωνικών Δικτύων Γράφων (GNN) για την ταξινόμηση αρχείων MIDI, στον τομέα της Ανάκτησης Μουσικής Πληροφορίας (Music Information Retrieval (MIR)). Τα δεδομένα MIDI, τύπος αναπαράστασης μουσικής πληροφορίας, μετατρέπονται σε δομές γράφων ώστε να καταγραφούν περίπλοκες εξαρτήσεις και σημασιολογικές σχέσεις. Αξιοποιώντας τεχνικές βαθιάς μάθησης, βασισμένες σε γράφους, στοχεύουμε στη βελτίωση της ακρίβειας και αποτελεσματικότητας της ταξινόμησης MIDI. Η διατριβή εξετάζει την ιστορία και τις προκλήσεις που συναντόνται στον τομέα του MIR, εστιάζοντας στην ανάλυση MIDI, και διερευνά διάφορες αρχιτεκτονικές GNN προσαρμοσμένες για δεδομένα MIDI. Μέσω των πειραμάτων, και της αξιολόγησης αυτών, αποδεικνύουμε την αποτελεσματικότητα των προτεινόμενων μεθοδολογιών, προσφέροντας ιδέες για την εφαρμογή μοντέλων βασισμένων σε γράφους για εργασίες μουσικής ταξινόμησης. Τα αποτελέσματα αυτής της έρευνας συμβάλλουν στον ευρύτερο τομέα του MIR, ενισχύοντας τις δυνατότητες των αυτοματοποιημένων συστημάτων κατανόησης και σύστασης μουσικής.

Λέξεις Κλειδιά. *Ετερογενείς γράφοι, Δεδομένα MIDI, Νευρωνικά δίκτυα γράφων, Ανάκτηση μουσικών πληροφοριών, Ταξινόμηση μουσικής, Ταξινόμηση μουσικών ειδών, Ταξινόμηση σε επίπεδο κόμβων, Βαθιά μάθηση για μουσική ανάλυση, Αναπαράσταση MIDI, Εξόρυξη μουσικών δεδομένων*

Abstract

This thesis explores the application of Heterogeneous Graphs and Graph Neural Networks (GNNs) for MIDI classification in the field of Music Information Retrieval (MIR). MIDI data, which represent musical notes and timing, are transformed into graph structures in order to capture dependencies and semantic relationships. By leveraging graph-based deep learning techniques, we aim to improve the accuracy and efficiency of MIDI classification. The thesis discusses the history and challenges of MIR, focusing on MIDI analysis, and investigates various GNN architectures tailored for MIDI data. Experimental evaluations demonstrate the effectiveness of the proposed methodologies, offering insights into the application of graph-based models for music classification tasks. The outcomes of this research contribute to the broader field of MIR, enhancing the capabilities of automated music understanding and recommendation systems.

Keywords. *Heterogeneous graphs, MIDI data, Graph neural networks, Music information retrieval, Music classification, Music genre classification, Node-level classification, Deep learning for music analysis, MIDI representation, Music data mining*

Ευχαριστίες

Αρχικά, θα ήθελα να ευχαριστήσω τους επιβλέποντες μου, κ.Συμβώνη και κ.Στάμου, για την εμπιστοσύνη, την καθοδήγηση και τη βοήθεια που μου προσέφεραν καθ΄ όλη τη διάρκεια της συγγραφής αυτής της εργασίας. Επιπλέον, ένα τεράστιο ευχαριστώ στον Σπύρο Κανταρέλη, που με βοήθησε να ανακαλύψω και να αναπτύξω το παρών θέμα μέσω της συνεργασίας μας, καθώς και στους Βασίλη Λυμπεράτο και Αγγελική Δημητρίου για τις συζητήσεις, τις ιδέες και τις συμβουλές που μοιράστηκαν μαζί μου. Τέλος, δεν θα μπορούσα να είχα φέρει εις πέρας αυτή την πρόκληση, εάν δεν είχα δίπλα μου τους κοντινούς μου ανθρώπους, την Ανδριανή, τον Αποστόλη, τον Δημήτρη και τον Χάρη. Τους ευχαριστώ για την αντοχή και την υποστήριξη.

Και ένα ακόμα ευχαριστώ στον γάτο μου Μίου, για τις ατελείωτες ώρες που πέρασε μαζί μου ενώ έγραφα, κρατώντας μου συντροφιά.

Περιεχόμενα

Κατάλογος Σχημάτων

Κατάλογος Πινάκων

Κεφάλαιο 1

Εισαγωγή

Η Ανάκτηση Μουσικής Πληροφορίας (Music Information Retrieval, MIR) είναι ένας δυναμικός τομέας στη συμβολή της μουσικής, της επεξεργασίας σήματος και της τεχνητής νοημοσύνης. Στόχος του είναι η ανάπτυξη αλγορίθμων και τεχνικών που μας επιτρέπουν να αναλύουμε, να κατανοούμε και να εξάγουμε σημαντικές πληροφορίες από τη μουσική σε διάφορες μορφές, συμπεριλαμβανομένων ηχογραφήσεων, μουσικών σημειώσεων και μεταδεδομένων. Τα τελευταία χρόνια, η MIR έχει γνωρίσει αξιοσημείωτες προόδους, χάρη στην έλευση της βαθιάς μάθησης και των μοντέλων που βασίζονται σε γράφους. Η παρούσα διατριβή επικεντρώνεται στην εφαρμογή των ετερογενών γράφων και των νευρωνικών δικτύων γράφων (Graph Neural Networks, GNN) στο πλαίσιο της ταξινόμησης MIDI, μιας σημαντικής εργασίας στο χώρο του MIR.

Το MIDI (Musical Instrument Digital Interface) είναι ένα ευρέως υιοθετημένο πρότυπο για την αναπαράσταση της μουσικής σε ψηφιακή μορφή. Κωδικοποιεί μουσικές νότες, χρονισμό και άλλες σχετικές πληροφορίες, καθιστώντας το πολύτιμο πόρο για την υπολογιστική μουσική ανάλυση. Η ταξινόμηση MIDI περιλαμβάνει την αυτόματη κατηγοριοποίηση αρχείων MIDI σε συγκεκριμένα είδη, στυλ, όργανα ή άλλες σημασιολογικές ετικέτες. Χρησιμεύει ως θεμελιώδες δομικό στοιχείο για εφαρμογές όπως η σύσταση μουσικής, η αναγνώριση ειδών και η παραγωγή μουσικής.

Η χρήση των ετερογενών γράφων και των νευρωνικών δικτύων γράφων στην ταξινόμηση MIDI προσφέρει πολλά υποσχόμενες ευκαιρίες για την αξιοποίηση της έμφυτης δομής και των σχέσεων εντός των δεδομένων MIDI. Με την αναπαράσταση των αρχείων MIDI ως γράφων και την εφαρμογή τεχνικών βαθιάς μάθησης γράφων, μπορούμε να συλλάβουμε τόσο τοπικές όσο και καθολικές εξαρτήσεις, να εκμεταλλευτούμε τις ιεραρχικές σχέσεις και να ενσωματώσουμε πλούσιες σημασιολογικές πληροφορίες για βελτιωμένη απόδοση ταξινόμησης.

Αξιοποιώντας τις πλούσιες πληροφορίες που είναι κωδικοποιημένες στα αρχεία MIDI, η παρούσα έρευνα συμβάλλει στον ευρύτερο τομέα του MIR και παρέχει πολύτιμες πληροφορίες για την εφαρμογή των Ετερογενών Γράφων και των Νευρωνικών Δικτύων Γράφων για εργασίες ταξινόμησης μουσικής. Τα ευρήματα και οι μεθοδολογίες που αναπτύχθηκαν σε αυτή τη διατριβή μπορούν ενδεχομένως να βελτιώσουν την ακρίβεια και την αποτελεσματικότητα των συστημάτων ταξινόμησης MIDI, ανοίγοντας νέους

δρόμους για τη μουσική ανάλυση και την αυτοματοποιημένη κατανόηση της μουσικής.

Στα επόμενα κεφάλαια, θα εμβαθύνουμε στις λεπτομέρειες της αναπαράστασης δεδομένων MIDI, της κατασκευής γράφων, των αρχιτεκτονικών νευρωνικών δικτύων γράφων, των πειραματικών μεθοδολογιών και των μετρικών αξιολόγησης. Θα παρουσιάσουμε και θα αναλύσουμε τα αποτελέσματα που προέκυψαν από διάφορα πειράματα που διεξήχθησαν σε διάφορα σύνολα δεδομένων MIDI. Τέλος, θα συζητήσουμε τις επιπτώσεις αυτής της έρευνας, τους περιορισμούς της και τις πιθανές μελλοντικές κατευθύνσεις στον τομέα της μουσικής ταξινόμησης.

Η παρούσα διατριβή σκοπεύει να στηριχθεί στην υφιστάμενη βιβλιογραφία, και κυρίως στην [41], η οποία αποτέλεσε το κίνητρο για την προσέγγιση που παρουσιάζουμε. Στοχεύουμε στο να να συνεισφέρουμε στον αυξανόμενο όγκο πληροφοριών στον τομέα της Ανάκτησης Μουσικής Πληροφορίας, και να παρέχουμε πρακτικές γνώσεις σε ερευνητές, επαγγελματίες και ενθουσιώδεις χρήστες που ενδιαφέρονται για την ανάλυση και ταξινόμηση μουσικών αρχείων MIDI.

Κεφάλαιο 2

Θεωρία Γράφων

2.1 Εισαγωγή

Τα γραφήματα είναι ισχυρά εργαλεία για τη μοντελοποίηση πολύπλοκων σχέσεων και εξαρτήσεων μεταξύ οντοτήτων, γεγονός που τα καθιστά φυσικά κατάλληλα για πολλές εφαρμογές με πραγματικά δεδομένα, όπως η επιστήμη των υπολογιστών, η επιχειρησιακή έρευνα, οι κοινωνικές επιστήμες, η βιολογία και η φυσική. Πιο συγκεκριμένα, τα τελευταία χρόνια έχουν αποκτήσει όλο και μεγαλύτερη σημασία στον τομέα της μηχανικής μάθησης, με εφαρμογές σε ένα ευρύ φάσμα τομέων όπως η αναγνώριση εικόνων, η επεξεργασία φυσικής γλώσσας και τα συστήματα συστάσεων.

Εστιάζοντας στο θέμα της μηχανικής μάθησης, οι γράφοι μπορούν να χρησιμοποιηθούν για την αναπαράσταση δεδομένων με διάφορους τρόπους, είτε πρόκειται για ένα δίκτυο οντοτήτων και των σχέσεών τους, είτε για μια συλλογή αλληλένδετων αντικειμένων, είτε για ένα σύνολο κόμβων και ακμών σε μια δομή γράφου. Τα μοντέλα μηχανικής μάθησης που αξιοποιούν τις δομές γράφων μπορούν να μάθουν μοτίβα και σχέσεις μεταξύ κόμβων, ακμών και υπογράφων, επιτρέποντάς τους να κάνουν προβλέψεις και να εκτελούν εργασίες όπως η ομαδοποίηση, η ταξινόμηση και η πρόβλεψη.

Προσεγγίσεις που βασίζονται σε γράφους στη μηχανική μάθηση οδήγησε στην ανάπτυξη μιας ποικιλίας αλγορίθμων και τεχνικών, όπως τα νευρωνικά δίκτυα γράφων, οι εμφύτευση γράφων[1] και τα συνελικτικά δίκτυα γράφων. Οι τεχνικές αυτές έχουν αποδειχθεί ιδιαίτερα αποτελεσματικές για ένα ευρύ φάσμα εφαρμογών, και έχουν ανοίξει νέους δρόμους για την πρόοδο της έρευνας.

Σε αυτό το κεφάλαιο, θα εξερευνήσουμε τις θεμελιώδεις έννοιες και ιδιότητες των γράφων, καθώς και κάποιες ειδικές δομές που θα μας φανούν χρήσιμες στις εφαρμογές στη μηχανική μάθηση. Σε επόμενα κεφάλαια, θα δούμε πώς αυτές οι έννοιες μπορούν να εφαρμοστούν στο ερευνητικό πλαίσιο που δημιουργήθηκε από τους [41] και χρησιμοποιήθηκε στον τομέα της μουσικής ταξινόμησης.

2.2 Βασικά Θεωρίας Γράφων

2.2.1 Απλός μη-κατευθυνόμενος Γράφος

Στην απλούστερη μορφή του [5, σελ 148], ένας γράφος (ή γράφημα) μπορεί να αναπαρασταθεί ως ένα διατεταγμένο ζεύγος [2] $\mathcal{G} = (V, E)$, όπου:

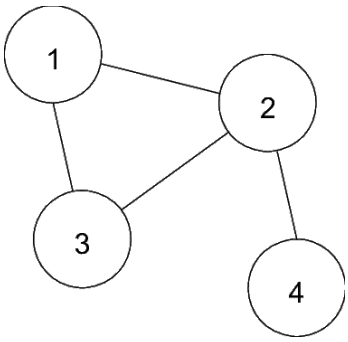
- V είναι ένα πεπερασμένο σύνολο κόμβων (γνωστοί και ως κορυφές),
- $E \subseteq \{ (u, v) \mid u, v \in V \ , \ u \neq v\}$, το οποίο είναι ένα πεπερασμένο σύνολο ακμών. Μια ακμή, (u, v), είναι ένα **μη-διατεταγμένο** [3] ζεύγος κόμβων.

[1] Η εμφύτευση αναφέρεται στη διαδικασία αναπαράστασης δεδομένων υψηλής διάστασης σε χώρο χαμηλότερης διάστασης, διατηρώντας τις βασικές τους ιδιότητες.

[2] ένα ζεύγος θεωρείται διατεταγμένο εάν $(a, b) \neq (b, a)$

[3] αντίστροφα, ένα ζεύγος θεωρείται μη-διατεταγμένο αν $(a, b) = (b, a)$

Ο $\mathcal{G}$ είναι επίσημα γνωστός ως απλός μη-κατευθυνόμενος γράφος, αλλά για συντομία θα τον αποκαλούμε γράφο[4]. Παρακάτω, μπορούμε να δούμε την εικόνα ενός γράφου με κόμβους $\{1, 2, 3, 4\}$.



Σχήμα 2.1: Ένας απλός μη-κατευθυνόμενος γράφος. Πηγή: [44]

2.2.2 Απλός κατευθυνόμενος Γράφος

Όπως υποδηλώνει και το όνομά του, ένας απλός κατευθυνόμενος γράφος, (ή εν συντομία κατευθυνόμενος γράφος), είναι ένας γράφος του οποίου οι ακμές έχουν προσανατολισμό. Δηλαδή, η ακμή (u, v) δείχνει από τον κόμβο u προς τον κόμβο v, ενώ η ακμή (v, u) είναι διαφορετική και δείχνει από τον v στον u.

Ορίζεται[5, σελ. 161], ως ένα διατεταγμένο ζεύγος $\mathcal{G} = (V, E)$, όπου:

- V είναι ένα πεπερασμένο σύνολο κόμβων,
- $E \subseteq \{ (u, v) \mid (u, v) \in V^2 \ , \ u \neq v\}$, είναι ένα πεπερασμένο σύνολο ακμών (επίσης γνωστές ως κατευθυνόμενες ακμές ή βέλη). Μια κατευθυνόμενη ακμή, (u, v), είναι ένα διατεταγμένο ζεύγος κόμβων.

Εδώ, το V^2 δηλώνει το καρτεσιανό γινόμενο $V \times V$, το οποίο είναι το σύνολο όλων των διατεταγμένων ζευγών (u, v), όπου $u, v \in V$. Η ακμή (v, u) ονομάζεται ανεστραμμένη ακμή της (u, v).

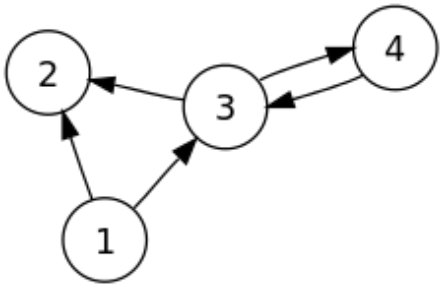
Στο σχήμα 2.2 μπορούμε να δούμε έναν κατευθυνόμενο γράφο.

Ένας *βρόγχος* είναι ένα ειδικό είδος ακμής, που συνδέει έναν κόμβο με τον εαυτό του, δηλαδή (u, u), $u \in V$. Ένα απλό γράφημα, όπως το ορίζουμε παραπάνω, δεν περιέχει βρόχους.

Ένας *σταθμισμένος γράφος* είναι ένας γράφος όπου κάθε ακμή συνδέεται με έναν πραγματικό αριθμό (το βάρος της). Τα βάρη μπορεί να αντιπροσωπεύουν το μήκος, το κόστος μετάβασης ή τις πιθανότητες μετάβασης.

[4] οι γράφοι αποκαλούνται επίσης και δίκτυα. Σε αυτό το κεφάλαιο, οι δυο αυτοί όροι θα χρησιμοποιούνται με την ίδια σημασία.

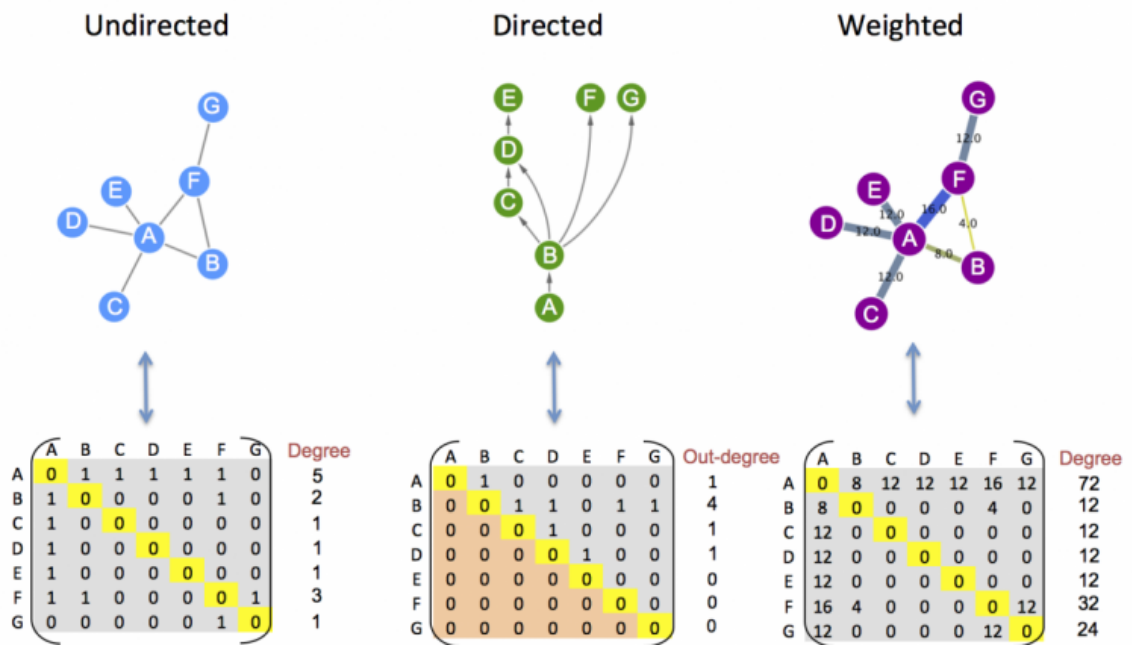
[5] https://commons.wikimedia.org/wiki/File:Directed_graph_no_background.svg



Σχήμα 2.2: Ένας απλός κατευθυνόμενος γράφος. Πηγή:[5]

2.3 Αναπαράσταση Γράφων

Υπάρχουν πολλοί τρόποι με τους οποίους μπορεί κανείς να αναπαραστήσει έναν γράφο. Σε αυτή την ενότητα, θα επικεντρωθούμε σε δύο από αυτές τις αναπαραστάσεις.



Σχήμα 2.3: Αναπαράσταση μέσω πινάκων γειτνίασης για διάφορα είδη γραφημάτων: η αριστερή εικόνα αναπαριστά τον πίνακα ενός μη-κατευθυνόμενου γράφου, η μεσαία ενός κατευθυνόμενου γράφου και η δεξιά ενός σταθμισμένου γράφου. Πηγή:[6]

Πίνακας Γειτνίασης: Ο πίνακας γειτνίασης[23] ενός γράφου είναι ένας πραγματικός τετραγωνικός πίνακας $\mathbf{A} \in \mathbb{R}^{|V|\times|V|}$ [7].

Για να κατασκευάσουμε τον πίνακα γειτνίασης, πρώτα επιλέγουμε ποιος κόμβος αντιπροσωπεύεται από ποια γραμμή και στήλη του πίνακα π.χ. ο κόμβος u είναι η i-οστή γραμμή i-οστή στήλη του $\mathbf{A}$, και ο κόμβος v η j-οστή. Δηλαδή, $\mathbf{A}[u, v] = a_{ij}$. Εάν $a_{ij} \neq 0$, τότε υπάρχει μια ακμή μεταξύ του κόμβου που έχει δείκτη i και του κόμβου που έχει δείκτη j, άρα $(u, v) \in E$. Ειδικότερα, εάν ο γράφος είναι ένας μη-σταθμισμένος γράφος, τότε $a_{ij} \in \{0, 1\}$. Αντίθετα, εάν ο γράφος έχει σταθμισμένες ακμές, τότε το

[6] https://i.pinimg.com/originals/c0/b6/16/c0b616689c89d4736767fbf30b5d2691.png
[7] $\mathbb{R}^{N\times M}$ αναφέρεται στο σύνολο όλων των πραγματικών πινάκων διάστασης $N \times M$.

στοιχείο a_{ij} μπορεί να λάβει οποιαδήποτε πραγματική τιμή, υποδεικνύοντας το βάρος της συγκεκριμένης ακμής. Επιπλέον, εάν ο γράφος είναι μη-κατευθυνόμενος, τότε ο $\mathbf{A}$ θα είναι συμμετρικός πίνακας, ενώ αν πρόκειται για κατευθυνόμενο γράφο, τότε δεν μπορούμε να εγγυηθούμε τη συμμετρία του $\mathbf{A}$. Ο πίνακας γειτνίασης μπορεί επίσης να χρησιμοποιηθεί για την κατασκευή άλλων χρήσιμων μορφών αναπαράστασης γράφων, τις οποίες θα εξερευνήσουμε σε επόμενα κεφάλαια.

Αναπαράσταση Edgelist: Όπως υποδηλώνει και το όνομά της, η αναπαράσταση edgelist (λίστα ακμών)[35] είναι μια λίστα που περιέχει τις ακμές του γράφου. Αυτή μπορεί να αναπαρασταθεί είτε με έναν πίνακα $|E| \times 2$, όπου κάθε γραμμή περιέχει τους κόμβους αρχής και τέλους της ακμής, είτε με μια λίστα από υπολίστες μήκους 2, με κάθε υπολίστα να είναι ο κόμβος αρχής και τέλους της ακμής. Αν δουλεύουμε με σταθμισμένες ακμές, κάθε γραμμή (ή υπολίστα) μπορεί να περιέχει έναν τρίτο αριθμό, συνήθως στο τέλος, ο οποίος δηλώνει το βάρος της. Η αναπαράσταση edgelist είναι ένας αποδοτικός τρόπος αποθήκευσης γραφημάτων σε αρχεία `.edgelist` (μια παραλλαγή του μοντέλου αρχείου `.csv`), όπου στη συνέχεια μπορούν εύκολα να φορτωθούν στη μνήμη.

2.3.1 Πίνακας Χαρακτηριστικών

Είναι επίσης δυνατό ένας γράφος να έχει χαρακτηριστικά που να αποδίδονται στους κόμβους (ή τις ακμές) του. Τις περισσότερες φορές, αυτά τα χαρακτηριστικά (ή γνωρίσματα) θα αντιπροσωπεύονται από κάποιον πραγματικό αριθμό. Κάθε κόμβος τότε θα έχει ένα αντίστοιχο διάνυσμα χαρακτηριστικών, έστω $\mathbf{x}_u \in \mathbb{R}^n$, για $u \in V$. Μπορούμε να ορίσουμε τον πίνακα χαρακτηριστικών του γραφήματος, ως έναν πίνακα $\mathbf{X} \in \mathbb{R}^{|V| \times n}$, όπου: $\mathbf{X} = [\mathbf{x}_{u_1}\, \mathbf{x}_{u_2}\, \ldots\, \mathbf{x}_{u_m}]^T$, όπου $|V| = m$.

Μερικά από αυτά τα χαρακτηριστικά μπορεί να είναι:

- *Βαθμός Κόμβου*: Ένα βασικό χαρακτηριστικό κόμβων είναι ο βαθμός, d_u, του κόμβου $u \in V$. Ο βαθμός d_u είναι ο αριθμός των ακμών που προσπίπτουν σε έναν κόμβο, ο οποίος δίνεται από τον τύπο[23]:

$$d_u = \sum_{v \in V} \mathbf{A}[u, v] \tag{2.1}$$

- *Κατευθυνόμενοι Βαθμοί Κόμβων*: Μια ειδική μορφή του βαθμού κόμβου εμφανίζεται όταν εργαζόμαστε με κατευθυνόμενους γράφους. Σε αυτή την περίπτωση, ορίζουμε δύο διαφορετικούς βαθμούς [53], τον βαθμό εξόδου d_u^{out}, και τον βαθμό εισόδου d_u^{in}. Ο βαθμός εξόδου μας δίνει τον αριθμό των ακμών που εξέρχονται από τον κόμβο u:

$$d_u^{out} = \sum_{v \in V} \mathbf{A}[u, v] \tag{2.2}$$

  και ο βαθμός εισόδου μας δίνει τον αριθμό των ακμών που εισέρχονται στον κόμβο u:

$$d_u^{in} = \sum_{v \in V} \mathbf{A}[v, u] \tag{2.3}$$

Ο συνολικός βαθμός του κόμβου u, ορίζεται ως:

$$d_u^{tot} = d_u^{out} + d_u^{in} \tag{2.4}$$

- Ένα χαρακτηριστικό μπορεί επίσης να είναι ειδικό σε κάθε περίπτωση. Για παράδειγμα, όταν μιλάμε για ένα κοινωνικό δίκτυο, ένα χαρακτηριστικό κόμβου μπορεί να είναι οποιαδήποτε πληροφορία σχετικά με τον χρήστη που αντιπροσωπεύεται από τον κόμβο. Όταν μιλάμε για τον Παγκόσμιο Ιστό (WWW), ο οποίος είναι επίσης ένας γράφος που συνδέει ιστότοπους, οι ίδιοι οι ιστότοποι είναι οι κόμβοι και τα χαρακτηριστικά των κόμβων μπορεί να είναι χαρακτηριστικά του συγκεκριμένου ιστότοπου.

2.4 Ετερογενείς Γράφοι

Έχει ήδη τεκμηριωθεί ότι τα γραφήματα είναι ένα πολύ βολικό εργαλείο για την αναπαράσταση δεδομένων και των συνδέσεών τους. Από τον προαναφερθέν Παγκόσμιο Ιστό, τα κοινωνικά δίκτυα, τα δίκτυα κυκλοφορίας και τα δίκτυα ηλεκτρικής ενέργειας, μπορούμε να δούμε ότι σχεδόν οποιαδήποτε δομή μπορεί να αναπαρασταθεί με μια αρχιτεκτονική που μοιάζει με γράφο. Μέχρι αυτό το σημείο όμως, δεν κάναμε διάκριση μεταξύ του είδους των κόμβων (ή ακμών) που μπορεί να έχει ένας τέτοιος γράφος. Υποθέσαμε ότι όλα τα στοιχεία των V και E ανήκουν στις ίδιες κατηγορίες. Αυτοί οι τύποι γράφων ονομάζονται *ομογενείς γράφοι*. Εύκολα μπορεί κανείς να διαπιστώσει ότι αυτό το είδος μοντέλου έχει τους περιορισμούς του για το πως μπορεί να μοντελοποιήσει δεδομένα διαφορετικών τύπων.

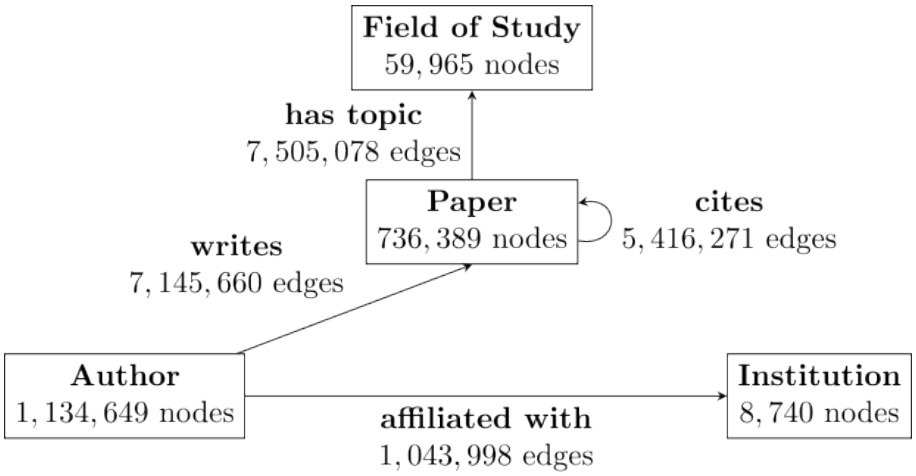
Για να παρουσιάσουμε την ανάγκη για διαφορετικούς τύπους κόμβων και ακμών, ας εξετάσουμε τη δομή του Facebook. Ακολουθώντας το παράδειγμα στο [68], μπορούμε να μοντελοποιήσουμε τις συνδέσεις του ως γράφο που αποτελείται από διαφορετικές κατηγορίες κόμβων. Αυτές οι κατηγορίες μπορεί να είναι χρήστες, επαγγελματικές σελίδες, αναρτήσεις, εικόνες και βίντεο. Οι συνδέσεις μεταξύ αυτών των κόμβων είναι επίσης ποικίλες. Υπάρχουν φιλίες χρηστών (user-friend-user), χρήστες που κάνουν like ή δημοσιεύουν δημοσιεύσεις (user- likes-post, userpublish- post), εικόνες ενσωματωμένες σε δημοσιεύσεις (picture-in-post), σελίδες που δημοσιεύουν εικόνες ή βίντεο (page-posting-picture, page-posting-video) κ.λπ.

Ακόμα ένα γνωστό παράδειγμα στη βιβλιογραφία, όπως παρουσιάζεται στο [31] (ή με ελαφρώς διαφορετική θεμελίωση στο [73]), είναι ένα ακαδημαϊκό δίκτυο. Οι τέσσερις τύποι κόμβων είναι paper, author, institution, και field of study (field). Οι συνδέσεις μεταξύ αυτών των κόμβων είναι author-affiliated_with-institution, author-writes-paper, paper-cites-paper και paper-has_topic-field.

Μπορούμε τώρα να προχωρήσουμε και να ορίσουμε επίσημα τον ετερογενή γράφο:

Ένας ετερογενής γράφος (H)[73] είναι ένας κατευθυνόμενος γράφος $\mathcal{H} = (V, E)$, εφοδιασμένος με 2 συναρτήσεις αντιστοίχισης:

[8] https://pytorch-geometric.readthedocs.io/en/latest/_images/hg_example.svg



Σχήμα 2.4: Το ακαδημαϊκό δίκτυο. Πηγή:[8]

- $\phi(u) : V \mapsto \mathcal{R}$, η οποία αντιστοιχίζει τους κόμβους στην αντίστοιχη κατηγορία κόμβων τους,
- $\varphi(e) : E \mapsto \mathcal{A}$, η οποία αντιστοιχίζει τις ακμές στην αντίστοιχη κατηγορία ακμών τους.

Το $\mathcal{R}$ είναι το σύνολο όλων των κατηγοριών κόμβων και $\mathcal{A}$ είναι το σύνολο όλων των κατηγοριών ακμών. Για να θεωρηθεί ο γράφος ετερογενής, απαιτούμε $|\mathcal{R}| + |\mathcal{A}| > 2$. Ορίζουμε το σχήμα του γράφου $\mathcal{H}$ (*graph schema*) ως $\mathcal{S}_{\mathcal{H}} = (\mathcal{R}, \mathcal{A})$. Το σχήμα $\mathcal{S}_{\mathcal{H}}$ είναι επίσης ένας γράφος που ορίζεται πάνω στις κατηγορίες κόμβων στο $\mathcal{R}$, με ακμές τις κατηγορίες ακμών στο $\mathcal{A}$.

Κεφάλαιο 3

Μηχανική μάθηση & Βαθιά μάθηση

3.1 Εισαγωγή

Τα νευρωνικά δίκτυα[1] έχουν αναδειχθεί ως ένα ισχυρό εργαλείο για την επίλυση ενός μεγάλου φάσματος πολύπλοκων προβλημάτων, από την αναγνώριση εικόνας και ομιλίας έως την επεξεργασία φυσικής γλώσσας και την ανάλυση μουσικής. Σε αυτό το κεφάλαιο, θα δώσουμε μια επισκόπηση των θεωρητικών βάσεων των νευρωνικών δικτύων, με σκοπό τη γενίκευση στα νευρωνικά δίκτυα γράφων, τα οποία θα παρουσιαστούν στο επόμενο κεφάλαιο.

Θα ξεκινήσουμε με την εισαγωγή στις βασικές έννοιες της μηχανικής μάθησης και των νευρωνικών δικτύων, συμπεριλαμβανομένης της ιστορίας τους και του τρόπου λειτουργίας τους. Στη συνέχεια, θα συζητήσουμε τους κύριους τύπους νευρωνικών δικτύων, όπως τα νευρωνικά δίκτυα πρόσθιας τροφοδότησης, τα συνελικτικά και τα αναδρομικά νευρωνικά δίκτυα, ενώ θα εξηγήσουμε επίσης τον τρόπο με τον οποίο χρησιμοποιούνται σε διάφορες εφαρμογές.

Θα συζητήσουμε επίσης τη διαδικασία εκπαίδευσης των νευρωνικών δικτύων, συμπεριλαμβανομένης του αλγορίθμου οπισθοδιάδοσης(backpropagation), των μεθόδων βελτιστοποίησης και των τεχνικών κανονικοποίησης. Στη συνέχεια, θα καλύψουμε διάφορες μεθόδους αξιολόγησης των νευρωνικών δικτύων, όπως το cross-validation και η δοκιμή σε αθέατα δεδομένα.

Συνολικά, το παρόν κεφάλαιο αποσκοπεί στην παροχή μιας ολοκληρωμένης επισκόπησης των θεωρητικών θεμελίων των νευρωνικών δικτύων και την χρήση τους για την ανάπτυξη πρακτικών εφαρμογών.

3.2 Εισαγωγή στη Μηχανική Μάθηση

Η μηχανική μάθηση (ML) είναι ένα υποπεδίο της τεχνητής νοημοσύνης που έχει ως στόχο να επιτρέψει στους υπολογιστές να μαθαίνουν από δεδομένα, και να κάνουν προβλέψεις ή να παίρνουν αποφάσεις χωρίς να ρητά προγραμματισμένοι για αυτές. Παρακάτω, θα δώσουμε μια σύντομη εισαγωγή στις θεμελιώδεις έννοιες της μηχανικής μάθησης, συμπεριλαμβανομένων των διαφορετικών τύπων εργασιών μηχανικής μάθησης και των συναρτήσεων κόστους.

3.2.1 Κατηγορίες Μηχανικής Μάθησης

Ένας τρόπος για να κατηγοριοποιήσουμε τους αλγορίθμους μηχανικής μάθησης είναι με βάση τον τύπο των δεδομένων στα οποία εκπαιδεύονται και τη φύση της διαδικασίας μάθησης. Σε αυτή την ενότητα, θα δώσουμε μια επισκόπηση των τριών κύριων προτύπων της μηχανικής μάθησης: την *επιβλεπόμενη μάθηση*, *μη επιβλεπόμενη μάθηση*, και *ενισχυτική μάθηση*. Θα εξηγήσουμε τις βασικές διαφορές μεταξύ αυτών των κα-

[1] Στο κεφάλαιο 2, εξηγήσαμε ότι ένας γράφος μπορεί επίσης να ονομαστεί δίκτυο. Από εδώ και στο εξής, όταν αναφέρουμε κάποιο δίκτυο, μιλάμε για το εν λόγω νευρωνικό δίκτυο, εκτός αν αναφέρεται διαφορετικά.

τηγοριών και θα δώσουμε μερικά παραδείγματα των τύπων προβλημάτων για τα οποία κάθε κατηγορία είναι κατάλληλη. Θα αναφερθούμε επίσης, εν συντομία, σε ορισμένες από τις υποκατηγορίες, οι οποίες βασίζονται σε διαφορετικές προσεγγίσεις ή τεχνικές εντός κάθε κατηγορίας ή ως συνδυασμός μιας ή περισσότερων κατηγοριών. Οι πληροφορίες που ακολουθούν προέρχονται κυρίως από το [6].

Οι τρεις βασικές κατηγορίες είναι οι εξής:

- **Επιβλεπόμενη μάθηση**: Στην επιβλεπόμενη μάθηση, ο αλγόριθμος μάθησης εκπαιδεύεται σε δεδομένα με ¨ετικέτες¨, όπου κάθε παράδειγμα στο σύνολο εκπαίδευσης x συνδυάζεται με μια αντίστοιχη μεταβλητή-στόχο y. Δηλαδή, για κάποιο $x \in X$, υπάρχει ένα $y \in Y$, τέτοιο ώστε (x, y). Ο στόχος του αλγορίθμου είναι να μάθει μια αντιστοίχηση, $h : X \mapsto Y$, μεταξύ των μεταβλητών εισόδου X και των μεταβλητών-στόχου Y, έτσι ώστε να μπορεί να κάνει ακριβείς προβλέψεις σε νέα, αθέατα δεδομένα. Συγκεκριμένα, έστω x, εκτός του συνόλου εκπαίδευσης, και ο αντίστοιχος στόχος y, ο αλγόριθμος προσπαθεί να υπολογίσει την τιμή $h(x)$ έτσι ώστε $L(h(x), y) \longrightarrow 0$, όπου $L(\cdot)$ είναι η συνάρτηση κόστους και θα παρουσιαστεί στο επόμενο κεφάλαιο. Μερικά κοινά παραδείγματα εργασιών επιβλεπόμενης μάθησης περιλαμβάνουν την ταξινόμηση, όπου η μεταβλητή-στόχος είναι μια κατηγορία, και τα δεδομένα εισόδου πρέπει να ταξινομηθούν σε μια από αυτές τις κατηγορίες, και η παλινδρόμηση, όπου η μεταβλητή-στόχος είναι μια συνεχής τιμή.
- **Μη επιβλεπόμενη μάθηση**: Στην μη επιβλεπόμενη μάθηση, ο αλγόριθμος εκπαιδεύεται σε δεδομένα εισόδου, $x \in X$, χωρίς αντίστοιχες μεταβλητές εξόδου. Αυτά τα δεδομένα τα ονομάζουμε μη χαρακτηρισμένα δεδομένα. Ο στόχος του αλγορίθμου είναι να βρει μοτίβα ή κρυφές δομές στα δεδομένα, όπως συστάδες ή λανθάνουσες μεταβλητές, χωρίς καμία προηγούμενη γνώση για το ποια μπορεί να είναι αυτά τα μοτίβα. Ορισμένα παραδείγματα εργασιών μάθησης χωρίς επίβλεψη περιλαμβάνουν την ομαδοποίηση, όπου ο στόχος είναι η ομαδοποίηση παρόμοιων σημείων δεδομένων, και τη μείωση διαστατικότητας, όπου ο στόχος είναι η εύρεση μιας αναπαράστασης των δεδομένων σε μικρότερες διαστάσεις.
- **Ενισχυτική Μάθηση**: Στην ενισχυτική μάθηση, ο αλγόριθμος μηχανικής μάθησης μαθαίνει αλληλεπιδρώντας με ένα περιβάλλον και λαμβάνοντας ανατροφοδότηση με τη μορφή ανταμοιβών ή ποινών. Ο στόχος του αλγορίθμου είναι να μάθει μια πολιτική ή μια συνάρτηση λήψης αποφάσεων που μεγιστοποιεί την αθροιστική ανταμοιβή με την πάροδο του χρόνου. Ορισμένα παραδείγματα περιλαμβάνουν τη ρομποτική, τα συστήματα ελέγχου ή εκμάθηση στρατηγικών παιγνίων.

Αν και τα περισσότερα προβλήματα εμπίπτουν γενικότερα σε μία από τις παραπάνω βασικές κατηγορίες, λόγω της ποικιλομορφίας της φύσης των προβλημάτων και των δεδομένων, υπήρχε ανάγκη για πιο συμπεριληπτικά πλαίσια. Ως εκ τούτου, γεννήθηκαν επεκτάσεις ή συνδυασμοί των παραπάνω κατηγοριών, για να καλυφθούν όλες οι πιθανές ανάγκες.

- **Ημιεπιβλεπόμενη μάθηση**: Η ημιεπιβλεπόμενη μάθηση συνδυάζει τις ιδέες της επιβλεπόμενης και μη επιβλεπόμενης μάθησης, χρησιμοποιώντας ένα μείγμα

δεδομένων με και χωρίς ετικέτες κατά τη διάρκεια της εκπαίδευσης. Αξιοποιεί τις πληροφορίες που υπάρχουν και στις δύο κατηγορίες για να βελτιώσει την απόδοση του μοντέλου, ειδικά όταν τα επισημασμένα δεδομένα είναι περιορισμένα ή είναι δύσκολο να αποκτηθούν.

- **Διαδικτυακή μάθηση (Online Learning)**: Η διαδικτυακή μάθηση είναι ένα πρότυπο μηχανικής μάθησης όπου το μοντέλο μαθαίνει από τα δεδομένα με διαδοχικό τρόπο, μία παρατήρηση τη φορά. Σε αντίθεση με την παραδοσιακή εκμάθηση σε παρτίδες, όπου το μοντέλο εκπαιδεύεται σε ένα σταθερό σύνολο δεδομένων, η διαδικτυακή μάθηση επιτρέπει στο μοντέλο να προσαρμόζεται και να ενημερώνει τις προβλέψεις του καθώς διατίθενται νέα δεδομένα. Μαθαίνει συνεχώς και επικαιροποιεί τη γνώση του, χωρίς να απαιτείται επανεκπαίδευση σε ολόκληρο το σύνολο δεδομένων. Αυτό καθιστά τη διαδικτυακή μάθηση κατάλληλη για σενάρια όπου τα δεδομένα παράγονται σε πραγματικό χρόνο ή καταφθάνουν σε συνεχή ροη.

- **Ενεργός Μάθηση (Active Learning)**: Η ενεργός μάθηση περιλαμβάνει την επαναληπτική επιλογή και επισήμανση των πιο πλούσιων σε πληροφορία περιπτώσεων από ένα μεγάλο σύνολο μη επισημασμένων δεδομένων. Στοχεύει στην ελαχιστοποίηση της ποσότητας των επισημασμένων δεδομένων που απαιτούνται για την εκπαίδευση, επιτυγχάνοντας παράλληλα υψηλή απόδοση στις προβλέψεις. Η επισήμανση δεδομένων μπορεί να είναι δαπανηρή και χρονοβόρα, ιδίως όταν πρόκειται για μεγάλες ποσότητες δεδομένων. Η ενεργός μάθηση αντιμετωπίζει αυτή την πρόκληση, επιλέγοντας ενεργά τα πιο κατατοπιστικά δείγματα προς επισήμανση, ενώ διατηρεί ή ακόμα και βελτιώνει την ακρίβεια του μοντέλου. Η βασική ιδέα είναι η αλληλεπίδραση ενός ενεργού εκπαιδευόμενου (ένα μοντέλο μηχανικής μάθησης) με έναν εκπαιδευτή (έναν ανθρώπινο σχολιαστή ή έναν αυτοματοποιημένο μηχανισμό επισήμανσης). Ο εκπαιδευόμενος επιλέγει ενεργά τις περιπτώσεις που θεωρεί πιο πολύτιμες ή αβέβαιες και ρωτά τον εκπαιδευτή για να λάβει τις ετικέτες τους. Οι νέες επισημασμένες περιπτώσεις προστίθενται στο σύνολο εκπαίδευσης και το μοντέλο επανεκπαιδεύεται για να βελτιώσει την απόδοσή του.

- **Transfer Learning**: Το πρότυπο Transfer learning λειτουργεί με την αξιοποίηση της γνώσης που αποκτήθηκε πάνω σε ένα πρόβλημα και την εφαρμογή της σε ένα άλλο παραπλήσιο έργο ή τομέα. Συμβάλλει στο να ξεπεραστεί η έλλειψη δεδομένων με ετικέτες στην παρούσα εργασία με τη χρήση προ-εκπαιδευμένων μοντέλων από κάποιο διαφορετικό αλλά συναφή πρόβλημα. Συνήθως, η προσέγγιση περιλαμβάνει την εκπαίδευση ενός μοντέλου με τη χρήση ενός μεγάλου συνόλου δεδομένων, ακολουθούμενη από τη χρήση αυτού του προ-εκπαιδευμένου μοντέλου στην εκπαίδευση ενός άλλου μοντέλου με στόχο την επίλυση του αρχικού προβλήματος.

3.2.2 Εργασίες μάθησης

Ένας άλλος τρόπος με τον οποίο μπορούμε να κατηγοριοποιήσουμε τους αλγορίθμους μηχανικής μάθησης, είναι η φύση του προβλήματος και η μορφή των δεδομένων εξόδου

(π.χ. συνεχής τιμή, διακριτή τιμή, δομημένα δεδομένα). Μερικά από τις πιο δημοφιλής εργασίες περιλαμβάνουν:

- **Παλινδρόμηση**: Οι εργασίες παλινδρόμησης περιλαμβάνουν την πρόβλεψη μιας συνεχούς ή αριθμητικής τιμής και εμπίπτουν στην κατηγορία της επιβλεπόμενης μάθησης. Παραδείγματα περιλαμβάνουν την πρόβλεψη της τιμής στις αγορές ακινήτων, των χρηματιστηριακών τιμών ή την εκτίμηση της ηλικίας ενός ατόμου με βάση διάφορα χαρακτηριστικά.

- **Ταξινόμηση**: Οι εργασίες ταξινόμησης περιλαμβάνουν την πρόβλεψη διακριτών ή κατηγορικών ετικετών και επίσης εμπίπτουν στην κατηγορία της επιβλεπόμενης μάθησης. Παραδείγματα περιλαμβάνουν την ταξινόμηση μηνυμάτων ηλεκτρονικού ταχυδρομείου ως σπαμ, την πρόβλεψη αν ένας πελάτης θα αποχωρήσει, ή την αναγνώριση διαφορετικών τύπων αντικειμένων σε μια εικόνα.

- **Συσταδοποίηση**: Οι εργασίες συσταδοποίησης περιλαμβάνουν την ομαδοποίηση παρόμοιων σημείων δεδομένων με βάση τα εγγενή μοτίβα ή τις ομοιότητές τους. Πρόκειται για μια εργασία μη επιβλεπόμενης μάθησης, όπου ο αλγόριθμος ανακαλύπτει αυτόματα την υποκείμενη δομή των δεδομένων. Ορισμένα κοινά παραδείγματα περιλαμβάνουν την τμηματοποίηση πελατών, την τμηματοποίηση εικόνων, την ανίχνευση ανωμαλιών και την ανάλυση κοινωνικών δικτύων.

- **Μείωση διαστατικότητας**: Η μείωση της διαστατικότητας αποσκοπεί στη μείωση του αριθμού των μεταβλητών εισόδου ή των χαρακτηριστικών διατηρώντας παράλληλα τις πιο σημαντικές πληροφορίες. Τεχνικές όπως η ανάλυση κύριων συνιστωσών (PCA)[33] και η t-SNE (t-Distributed Stochastic Neighbor Embedding)[43] χρησιμοποιούνται συνήθως για τη μείωση της διαστατικότητας.

- **Συστήματα συστάσεων**: Τα συστήματα συστάσεων χρησιμοποιούνται για την παροχή εξατομικευμένων συστάσεων στους χρήστες με βάση τις προτιμήσεις ή τη συμπεριφορά τους. Τα συστήματα αυτά χρησιμοποιούνται συνήθως στο ηλεκτρονικό εμπόριο, στις πλατφόρμες ροής και στα μέσα κοινωνικής δικτύωσης. Μπορεί να είναι ένας συνδυασμός επιβλεπόμενης και μη επιβλεπόμενης μάθησης ή ακόμη και ενισχυτικής μάθησης[76].

- **Ανάλυση χρονοσειρών**:Η ανάλυση χρονοσειρών περιλαμβάνει την ανάλυση και την πρόβλεψη σημείων δεδομένων που είναι χρονικός διατεταγμένα. Παραδείγματα περιλαμβάνουν την πρόβλεψη των τιμών των μετοχών, την πρόβλεψη του καιρού ή την πρόβλεψη της ζήτησης.

Το έργο της παρούσας εργασίας εμπίπτει στην κατηγορία της ταξινόμησης με ημιεπίβλεψη, όπου μόνο μία από τις κατηγορίες κόμβων του ετερογενούς γραφήματός μας έχει συσχετιστεί με μια ετικέτα.

3.2.3 Συνάρτηση Κόστους

Η συνάρτηση κόστους (επίσης γνωστή ως αντικειμενική συνάρτηση ή συνάρτηση απώλειας) είναι μια μαθηματική συνάρτηση $L : Y \times Y \mapsto \mathbb{R}$ που ποσοτικοποιεί την

απόκλιση μεταξύ της προβλεπόμενης εξόδου ενός μοντέλου μηχανικής μάθησης, $h(x_i)$, και της πραγματικής εξόδου (ή στόχου), y_i, που σχετίζεται με μια δεδομένη είσοδο.

Η συνάρτηση κόστους υπολογίζει πόσο καλά το μοντέλο αποδίδει σε μια συγκεκριμένη εργασία και χρησιμεύει ως οδηγός για τη διαδικασία μάθησης, επιτρέποντας στον αλγόριθμο βελτιστοποίησης να τροποποιεί τις παραμέτρους του μοντέλου ώστε να ελαχιστοποιεί τη διαφορά μεταξύ των προβλεπόμενων και των πραγματικών εξόδων. Μέσω της ελαχιστοποίησης της συνάρτησης κόστους, το μοντέλο προσπαθεί να βελτιώσει την ακρίβεια πρόβλεψης του και να συλλάβει τα υποκείμενα μοτίβα μέσα στα δεδομένα.

Η επιλογή μιας κατάλληλης συνάρτησης κόστους είναι μια κρίσιμη πτυχή της μηχανικής μάθησης και είναι ζωτικής σημασίας για αποτελεσματική εκπαίδευση των μοντέλων. Η επιλογή εξαρτάται από τη φύση του προβλήματος, τα χαρακτηριστικά των δεδομένων και την επιθυμητή συμπεριφορά του μοντέλου. Ορισμένες συναρτήσεις είναι πιο ευαίσθητες στις ακραίες τιμές, ενώ άλλες δίνουν προτεραιότητα σε ορισμένους τύπους σφαλμάτων έναντι άλλων. Είναι εξαιρετικά σημαντική η επιλογή μιας συνάρτησης κόστους που εναρμονίζεται με τους συγκεκριμένους στόχους και τα κριτήρια αξιολόγησης του προβλήματος.

Σε προβλήματα παλινδρόμησης, όπου ο στόχος είναι η πρόβλεψη μιας συνεχούς τιμής, οι πιο συνηθισμένες συναρτήσεις κόστους περιλαμβάνουν:

- *Μέσο Τετραγωνικό Σφάλμα(Mean Squared Error)* (ή L_2 *loss*):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - h(x_i))^2 \tag{3.1}$$

- *Τυπικό Σφάλμα (Root Mean Squared Error)*:

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - h(x_i))^2} \tag{3.2}$$

- *Μέσο Απόλυτο Σφάλμα (Mean Absolute Error)* (ή L_1 *loss*):

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - h(x_i)| \tag{3.3}$$

Αυτές οι συναρτήσεις μετρούν τη διαφορά μεταξύ των προβλεπόμενων και των πραγματικών τιμών. Το πλεονέκτημα του ΡΜΣΕ έναντι του ΜΣΕ είναι ότι, παίρνοντας την τετραγωνική ρίζα, λαμβάνουμε τιμές που βρίσκονται στην ίδια μονάδα με την αρχική μεταβλητή-στόχο. Αυτό διευκολύνει την ερμηνεία και τη σύγκριση της απόδοσης του μοντέλου. Το ΡΜΣΕ είναι ιδιαίτερα χρήσιμο όταν η κλίμακα της μεταβλητής-στόχου είναι σημαντική για την κατανόηση του μεγέθους των σφαλμάτων. Το MAE προτιμάται εάν υπάρχουν ακραίες τιμές στα δεδομένα.

Σε προβλήματα ταξινόμησης, όπου ο στόχος είναι η ανάθεση δεδομένων εισόδου σε προκαθορισμένες κλάσεις, χρησιμοποιούνται διαφορετικοί τύποι συναρτήσεων κόστους.

Η συνάρτηση *Cross-Entropy Loss* (ή αλλιώς *Αρνητική Λογαριθμική Πιθανοφάνεια* (*Negative Log Likelihood*) είναι η πιο συνηθισμένη επιλογή. Για ταξινόμηση πολλαπλών κλάσεων, με C στο πλήθος κλάσεις, ο τύπος είναι[21]:

$$CEL = -\frac{1}{n}\sum_{i=1}^{n}\sum_{c=1}^{C} y_i^{(k)} log(p_i^{(k)}) \tag{3.4}$$

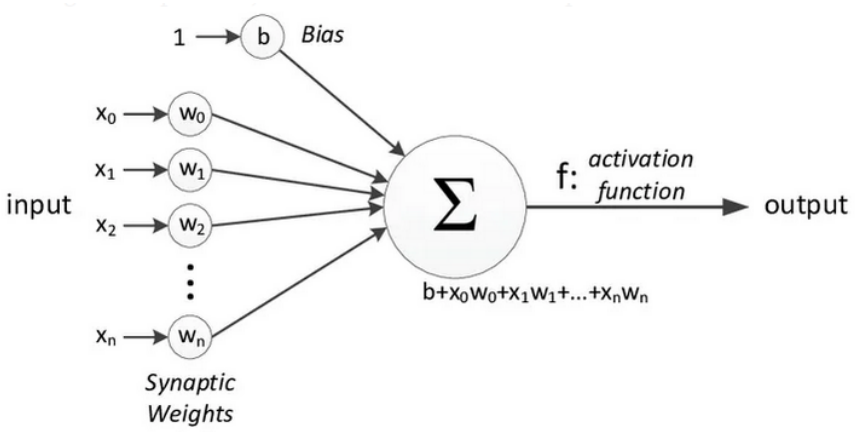
, όπου:

- $y_i^{(k)}$ είναι η πιθανότητα-στόχος εαν η περίπτωση i ανήκει στην κλάση k. Γενικά, είναι είτε ίση με 1 είτε με 0, ανάλογα με το αν η περίπτωση ανήκει στην κλάση ή όχι.
- $p_i^{(k)}$ είναι η προβλεπόμενη πιθανότητα ότι η περίπτωση i ανήκει στην κλάση k. Αυτή είναι και η έξοδος του μοντέλου μας.

3.3 Νευρωνικά Δίκτυα

Η βασική ιδέα πίσω από τα Τεχνητά Νευρωνικά Δίκτυα (Artificial Neural Networks (ANNs)), ή απλά Νευρωνικά Δίκτυα, είναι ο ανθρώπινος εγκέφαλος. Τα πρώτα σημάδια αυτής της έννοιας μπορούν να βρεθούν στο [46], όπου οι ερευνητές πρότειναν ένα μοντέλο με πηγή έμπνευσης τους ανθρώπινους νευρώνες, το οποίο θεωρητικά θα μπορούσε να υπολογίσει οποιαδήποτε λογική πρόταση. Έκτοτε, η έρευνα έχει επεκταθεί, γεννώντας πολλές διαφορετικές κατηγορίες ANNs.

Το πρώτο βήμα για τη μελέτη των νευρωνικών δικτύων είναι η μελέτη των δομικών τους στοιχείων, του perceptron.

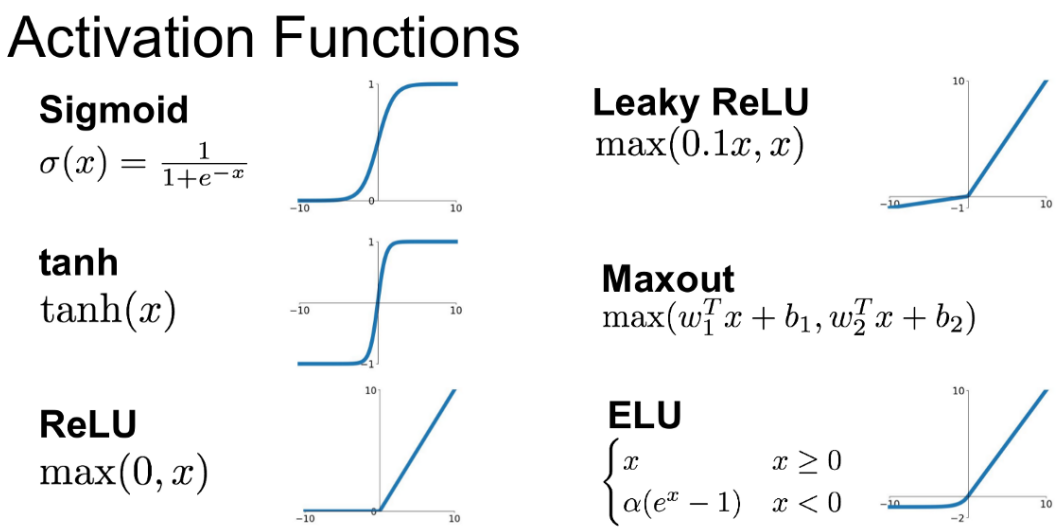


Σχήμα 3.1: Η αρχιτεκτονική του perceptron. Πηγή:[2]

[2] https://miro.medium.com/v2/resize:fit:552/0*eMcJxrc8Qmcm8Lhr.png

Το perceptron (ή γνωστό ως τεχνητός νευρώνας) είναι ένας γραμμικός συνδυασμός[3] του διανύσματος εισόδου x, που περνάει από μια **συνάρτηση ενεργοποίησης**. Τα βάρη μαθαίνονται κατά τη διάρκεια της εκπαίδευσης. Η συνάρτηση ενεργοποίησης ελέγχει την έξοδο του νευρώνα, καθορίζοντας εαν ο νευρώνας θα πρέπει να ενεργοποιηθεί ή όχι με βάση το σταθμισμένο άθροισμα. Μπορεί επίσης, αν επιλεγεί σωστά, να εισαγάγει μη γραμμικότητα στο μοντέλο μας. Αυτή ακριβώς είναι η συμπεριφορά που επιθυμούμε, καθώς θα του επιτρέψει να επιτύχει σε πιο σύνθετες (μη τετριμμένες) εργασίες χρησιμοποιώντας μόνο ένα μικρό αριθμό κόμβων[28].

Παρακάτω μπορούμε να δούμε μερικές από τις πιο δημοφιλείς επιλογές για συναρτήσεις ενεργοποίησης.



Σχήμα 3.2: Μερικές από τις πιο συνηθισμένες συναρτήσεις ενεργοποίησης, μαζί με τα γραφήματά τους. Πηγή: [4]

Η χρήση των συναρτήσεων ενεργοποίησης εξαρτάται σε μεγάλο βαθμό από τη φύση του προβλήματος, καθώς και από την αρχιτεκτονική του μοντέλου μας. Για παράδειγμα, η συνάρτηση softmax[12] (μια γενίκευση της λογιστικής συνάρτησης) χρησιμοποιείται κυρίως στο επίπεδο εξόδου των μοντέλων ταξινόμησης πολλαπλών κλάσεων. Λαμβάνει τις τιμές από το προηγούμενο στρώμα και τις κανονικοποιεί στην κατανομή πιθανότητας των κλάσεων εξόδου.

3.3.1 Το πρόβλημα της εξαφανιζόμενης κλίσης & η συνάρτηση ReLU

Πιθανώς η πιο ευρέως χρησιμοποιούμενη συνάρτηση ενεργοποίησης είναι η συνάρτηση ReLU και οι παραλλαγές της. Ένα σημαντικό πλεονέκτημα της ReLU είναι η ικα-

[3] Γραμμικός συνδυασμός δύο (ή περισσότερων) τιμών είναι το σταθμισμένο άθροισμά τους π.χ. ο γραμμικός συνδυασμός των x και y είναι $ax + by$, $a, b \in \mathbb{R}$

[4] https://cdn-images-1.medium.com/v2/resize:fit:1600/1*ZafDv3VUm60Eh10OeJu1vw.png

νότητά της να μετριάζει **το πρόβλημα της εξαφανιζόμενης κλίσης** (vanishing gradient problem)[4].

Το πρόβλημα της εξαφανιζόμενης κλίσης εμφανίζεται όταν το σήμα κλίσης που διαδίδεται προς τα πίσω μέσω των επιπέδων ενός βαθύ νευρωνικού δικτύου μειώνεται εκθετικά, καθιστώντας δύσκολη την αποτελεσματική μάθηση των προηγούμενων επιπέδων. Προκύπτει με συναρτήσεις ενεργοποίησης όπως η sigmoid ή η tanh, οι οποίες έχουν κορεσμένες περιοχές (δηλαδή περιοχές όπου η παράγωγος είναι πολύ κοντά στο μηδέν) για ακραίες τιμές εισόδου. Κατά τη διάρκεια της οπισθοδιάδοσης, οι κλίσεις συρρικνώνονται καθώς διέρχονται από αυτές τις κορεσμένες περιοχές, με αποτέλεσμα αδύναμες ενημερώσεις στα βάρη των προηγούμενων στρωμάτων και πιο αργή σύγκλιση.

Η ReLU αντιμετωπίζει αυτό το πρόβλημα αποφεύγοντας τον κορεσμό στη θετική περιοχή. Δεδομένου ότι η ReLU υφίσταται κορεσμό μόνο στην αρνητική πλευρά (όπου η παράγωγος είναι μηδέν), δεν πάσχει από κορεσμό κλίσης για θετικές εισόδους. Ως αποτέλεσμα, επιτρέπει στην αποτελεσματικότερη ροή κλίσης μέσω του δικτύου, δημιουργώντας καλύτερες ενημερώσεις βάρους και ταχύτερη σύγκλιση κατά τη διάρκεια της εκπαίδευσης.

Ωστόσο, η συνάρτηση ReLU δεν υπάρχει χωρίς τα προβλήματά της. Ένα αξιοσημείωτο μειονέκτημα είναι το πρόβλημα του 'dying ReLU¨, όπου ένας νευρώνας ReLU καθίσταται μόνιμα ανενεργός (με μηδενική έξοδο) λόγω των βαρών και των biases που προκαλούν τις εισόδους του να είναι αρνητικές για όλα τα παραδείγματα εκπαίδευσης. Σε τέτοιες περιπτώσεις, ο νευρώνας ουσιαστικά ¨πεθαίνει' και δεν συμβάλλει πλέον στη διαδικασία μάθησης. Διάφορες τροποποιήσεις της ReLU, όπως η Leaky ReLU και η Parametric ReLU (βλ. παραπάνω), έχουν προταθεί για τον μετριασμό αυτού του προβλήματος. Ένας άλλος τρόπος επίλυσης του συγκεκριμένου προβλήματος είναι ο έλεγχος της αρχικοποίησης των βαρών[42].

Συνοπτικά, η ReLU είναι μια αποτελεσματική συνάρτηση ενεργοποίησης που βοηθά στην αντιμετώπιση του προβλήματος της εξαφανιζόμενης κλίσης στα βαθιά νευρωνικά δίκτυα. Επιπλέον, η ικανότητά της να παρέχει αποδοτικό υπολογισμό, μη γραμμικότητα και να αποτρέπει τον κορεσμό για θετικές εισόδους, την έχει καταστήσει δημοφιλή επιλογή σε πολλά μοντέλα βαθιάς μάθησης τελευταίας τεχνολογίας.

Στη συνέχεια θα διερευνήσουμε πώς όντως εκπαιδεύουμε ένα δίκτυο, καθώς και ποιοι αλγόριθμοι και υπολογιστικές μέθοδοι χρησιμοποιούνται.

3.3.2 Αλγόριθμος Καθόδου Κλίσης (Gradient Descent)

Ο αλγόριθμος καθόδου κλίσης (ή απότομης καθόδου) είναι ένας επαναληπτικός αλγόριθμος βελτιστοποίησης που χρησιμοποιείται για την ελαχιστοποίηση της συνάρτησης κόστους ενός μοντέλου μηχανικής μάθησης. Είναι μια θεμελιώδης τεχνική στις μεθόδους μάθησης που βασίζονται στην κλίση, οι οποίες στηρίζονται στον υπολογισμό και την ενημέρωση των κλίσεων της συνάρτησης απώλειας σε σχέση με τις παραμέτρους του μοντέλου.

Ο στόχος του αλγορίθμου καθόδου κλίσης είναι να βρεθεί το σύνολο των παραμέτρων του μοντέλου (βάρη κλπ.) που ελαχιστοποιούν τη συνάρτηση κόστους, βελτιώνοντας

έτσι την απόδοση του μοντέλου. Λειτουργεί με επαναληπτική ενημέρωση των παραμέτρων προς την κατεύθυνση της πιο απότομης καθόδου της συνάρτησης κόστους.

Ο επαναληπτικός τύπος για τον υπολογισμό των βαρών έχει ως εξής:

$$\mathbf{w}^{(n+1)} = \mathbf{w}^{(n)} - \gamma_n \nabla_{\mathbf{w}} L(\mathbf{w}^{(n)}) \tag{3.5}$$

, όπου:

- $\mathbf{w}^{(n)}$ είναι το διάνυσμα των παραμέτρων στην επανάληψη n
- $\nabla_{\mathbf{w}} L(\mathbf{w})$ είναι η κλίση της συνάρτησης κόστους L ως προς τις παραμέτρους του μοντέλου
- γ είναι μια ρυθμιστική παράμετρος που ονομάζεται *ρυθμός μάθησης* (learning rate). Στο παρελθόν, ο ρυθμός μάθησης επιλέγονταν δοκιμάζοντας διαφορετικές τιμές και διατηρώντας αυτή με την καλύτερη απόδοση. Νεότερες μέθοδοι αλλάζουν ενεργά τον ρυθμό κατά τη διάρκεια της εκπαίδευσης, προκειμένου να ενισχύσουν την απόδοση και την ταχύτητα σύγκλισης. Ορισμένες από αυτές τις μεθόδους είναι ο προγραμματισμός (scheduling)[54], οι κυκλικοί ρυθμοί μάθησης (cyclical learning rates)[67] και οι προσαρμοστικές μέθοδοι (adaptive methods, θα παρουσιαστούν παρακάτω).

Η γενική διαδικασία της καθόδου κλίσης περιλαμβάνει τα ακόλουθα βήματα:

1. **Αρχικοποίηση**: Αρχικοποίηση των παραμέτρων του μοντέλου (βάρη και biases) με κάποιες αρχικές τιμές: είτε τυχαίες, με βάση μια καθορισμένη κατανομή[8] ή από άλλο παρόμοιο μοντέλο, π.χ. transfer learning[2].
2. **Εμπρόσθιο πέρασμα:**: Εκτέλεση ενός εμπρόσθιου περάσματος μέσω του μοντέλου για τον υπολογισμό της προβλεπόμενης εξόδου για μια δεδομένη είσοδο.
3. **Υπολογισμός Κόστους**: Υπολογισμός της συνάρτησης κόστους, η οποία ποσοτικοποιεί τη διαφορά μεταξύ της προβλεπόμενης εξόδου και της πραγματικής εξόδου.
4. **Οπισθοδιάδοση (Backpropagation)**: Στο βήμα της οπισθοδιάδοσης[62], υπολογίζονται οι κλίσεις της συνάρτησης κόστους σε σχέση με τις παραμέτρους του μοντέλου. Ο υπολογισμός ξεκινά από το επίπεδο εξόδου και κινείται προς τα πίσω μέσω του δικτύου.

   (α′) *Υπολογισμός κλίσης για το στρώμα εξόδου*: Οι κλίσεις της συνάρτησης κόστους υπολογίζονται σε σχέση με τις παραμέτρους στο στρώμα εξόδου με χρήση του κανόνα της αλυσίδας του διαφορικού λογισμού. Οι κλίσεις δείχνουν πόσο θα αλλάξει η τιμή της συνάρτησης κόστους για μια μικρή αλλαγή τις παραμέτρους.

(β′) *Υπολογισμός κλίσης για κρυφά στρώματα*: Στη συνέχεια, οι κλίσεις υπολογίζονται επαναληπτικά για τα προηγούμενα στρώματα, διαδίδοντας τις κλίσεις προς τα πίσω μέσα στο δίκτυο. Αυτό γίνεται με την εφαρμογή του κανόνα της αλυσίδας για τον υπολογισμό των κλίσεων σε σχέση με τις παραμέτρους σε κάθε στρώμα.

5. **Ενημέρωση παραμέτρων**: Ενημέρωση των παραμέτρων του μοντέλου κάνοντας ένα βήμα προς την αντίθετη κατεύθυνση των κλίσεων. Το μέγεθος κάθε βήματος, ο ρυθμός μάθησης, καθορίζει το μέγεθος της ενημέρωσης.

6. **Επανάληψη μέχρι τη σύγκλιση**: Επανάληψη των βημάτων του εμπρόσθιου περάσματος, του υπολογισμού των απωλειών, του υπολογισμού της κλίσης και της ενημέρωσης των παραμέτρων μέχρι να συγκλίνει ο αλγόριθμος ή να ικανοποιηθεί ένα προκαθορισμένο κριτήριο διακοπής (π.χ. μέγιστος αριθμός επαναλήψεων ή επίτευξη ενός επιθυμητού επιπέδου ακρίβειας).

Στην πράξη, παραλλαγές της κάθοδος κλίσης, όπως η *στοχαστική κάθοδος κλίσης stochastic gradient descent (SGD)*[65] και *mini-batch gradient descent*[34], χρησιμοποιούνται συχνά για τον αποτελεσματικό χειρισμό μεγάλων συνόλων δεδομένων. Αυτές οι παραλλαγές ενημερώνουν τις παραμέτρους χρησιμοποιώντας υποσύνολα των δεδομένων εκπαίδευσης, γεγονός που μπορεί να επιταχύνει σημαντικά τη διαδικασία μάθησης.

Ορισμένες άλλες δημοφιλείς μέθοδοι, οι οποίες κάνουν χρήση προσαρμοστικών ρυθμών μάθησης, είναι: *AdaGrad*[17], *RMSProp*[71] και *Adam*[36].

3.3.3 Εκπαίδευση, Επαλήθευση και Δοκιμή

Ένα μέρος της εκπαίδευσης που δεν έχουμε θίξει μέχρι στιγμής είναι η προετοιμασία του συνόλου δεδομένων. Αντί να χρησιμοποιείται ολόκληρο το σύνολο για εκπαίδευση, η συνήθης πρακτική είναι να χωρίζεται σε 3 διαφορετικά σύνολα: το σύνολο εκπαίδευσης (training), το σύνολο επαλήθευσης (validation) και το σύνολο δοκιμής (test). Οι συνηθισμένες αναλογίες διαχωρισμού αυτών των συνόλων, ως προς το αρχικό σύνολο, είναι 65% training, 15% validation και 20% test. Παρακάτω, θα κάνουμε μια σύντομη παρουσίαση του ρόλου του κάθε υποσυνόλου στο κομμάτι της εκπαίδευσης:

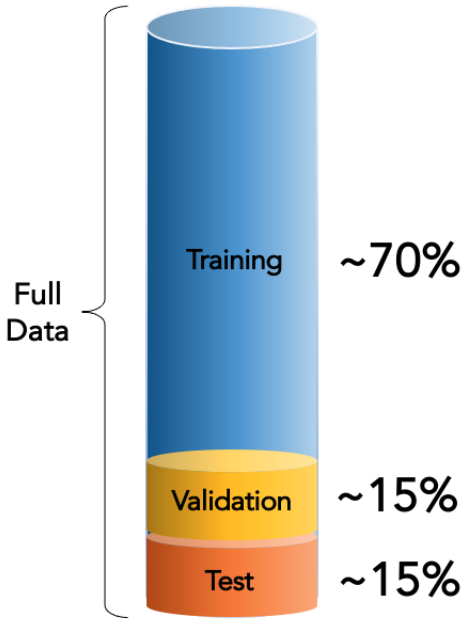
- **Σύνολο εκπαίδευσης**: Το σύνολο εκπαίδευσης, όπως υποδηλώνει και το όνομά του, αποτελείται από τα δεδομένα που χρησιμοποιούνται κατά την εκπαίδευση. Το μοντέλο χρησιμοποιεί αυτά τα δεδομένα για να βελτιώσει την απόδοσή του ελαχιστοποιώντας το κόστος εκπαίδευσης.

- **Σύνολο επαλήθευσης**: Το σύνολο επαλήθευσης χρησιμοποιείται για τη λεπτομερή ρύθμιση του μοντέλου κατά τη διάρκεια της διαδικασίας εκπαίδευσης και για τη λήψη αποφάσεων σχετικά με τη ρύθμιση των υπερπαραμέτρων[5]. Λειτουργεί ως υποκατάστατο για αθέατα δεδομένα και βοηθά στην αξιολόγηση της απόδοσης

[5]Οι υπερπαράμετροι είναι παράμετροι που δεν μαθαίνονται από το μοντέλο κατά τη διάρκεια της διαδικασίας εκπαίδευσης, αλλά καθορίζονται πριν από την εκπαίδευση, π.χ. βάθος του δικτύου

του μοντέλου σε παραδείγματα στα οποία δεν έχει εκπαιδευτεί άμεσα. Το σύνολο επαλήθευσης είναι ουσιώδες για την παρακολούθηση της προόδου του μοντέλου και την αποφυγή της υπερπροσαρμογής (θα συζητηθεί παρακάτω). Με βάση τις επιδόσεις κατά την επαλήθευση, μπορούν να γίνουν προσαρμογές στην αρχιτεκτονική του μοντέλου, στις τεχνικές κανονικοποίησης ή στις υπερπαραμέτρους για τη βελτιστοποίηση των επιδόσεών του.

- **Σύνολο Δοκιμής**: Μόλις ολοκληρωθούν οι φάσεις εκπαίδευσης και επαλήθευσης του μοντέλου, το τελικό μοντέλο αξιολογείται στο σύνολο δοκιμών. Το σύνολο δοκιμής παραμένει ανέγγιχτο κατά τη διάρκεια της διαδικασίας εκπαίδευσης, εξασφαλίζοντας μια αμερόληπτη αξιολόγηση της απόδοσης του μοντέλου. Υπολογίζονται οι μετρικές δοκιμής, παρέχοντας ένα αντικειμενικό μέτρο της ικανότητας του μοντέλου να γενικεύει σε νέα, αθέατα δεδομένα.



Σχήμα 3.3: Διαχωρισμός Train-Validation-Test. Πηγή: [6]

Μια ισχυρή τεχνική για την αξιοποίηση των πλεονεκτημάτων αυτού του διαχωρισμού, είναι το επονομαζόμενο *cross-validation*. Στην αρχή, το αρχικό υποσύνολο εκπαίδευσης και επαλήθευσης ανακατεύεται και χωρίζεται σε k ίσα υποσύνολα (k-folds). Σε κάθε βήμα της διαδικασίας, 1 από αυτά τα υποσύνολα χρησιμοποιείται ως σύνολο επαλήθευσης και τα υπόλοιπα $k-1$ ως σύνολο εκπαίδευσης. Η μέθοδος αυτή είναι γνωστή ως k-fold cross-validation[39]. Άλλες παραλλαγές αυτής της μεθόδου είναι η stratified k-fold cross-validation, η οποία διατηρεί την κατανομή των κλάσεων σε κάθε υποσύνολο και η leave-one-out cross-validation, όπου κάθε σημείο στο σύνολο δεδομένων χρησιμεύει ως ξεχωριστό σύνολο επαλήθευσης. Και οι δύο αυτές μέθοδοι παρουσιάζονται επίσης στο [39].

[6] https://thaddeus-segura.com/wp-content/uploads/2021/06/Screen-Shot-2021-06-17-at-7.03.33-PM-1.png

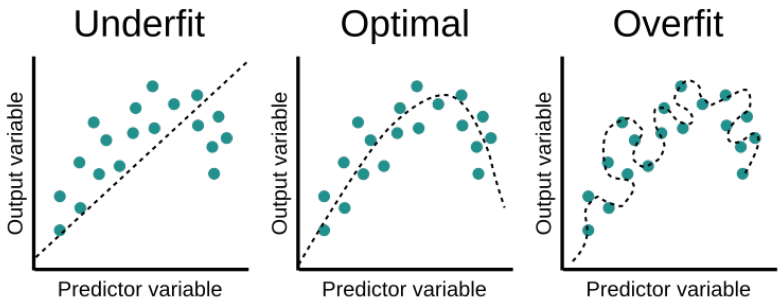
Με την επανειλημμένη εκπαίδευση και αξιολόγηση του μοντέλου σε διαφορετικά υποσύνολα, η μέθοδος cross-validation παρέχει μια πιο αξιόπιστη εκτίμηση της απόδοσης του μοντέλου σε σχέση με ένα μόνο σύνολο επαλήθευσης, ενώ παράλληλα μειώνει τη διακύμανση του μοντέλου. Αυτό αποδεικνύεται εξαιρετικά πολύτιμο εργαλείο για την επιλογή του σωστού μοντέλου, των υπερπαραμέτρων αυτού, καθώς και για τη λήψη τεκμηριωμένων αποφάσεων σχετικά με τη δυνατότητα γενίκευσης του μοντέλου.

3.3.4 Υποπροσαρμογή & Υπερπροσαρμογή

Εκτός από το πρόβλημα της εξαφανιζόμενης κλίσης που ήδη συζητήσαμε, δύο άλλες αρκετά συνηθισμένες προκλήσεις που αντιμετωπίζονται κατά την εκπαίδευση είναι η *υποπροσαρμογή* (underfitting), και η *υπερπροσαρμογή* (overfitting). Το καθένα από αυτά τα προβλήματα σχετίζεται με την ικανότητα του μοντέλου να γενικεύει τις γνώσεις του σε νέα, αθέατα δεδομένα.

Η υποπροσαρμογή συμβαίνει όταν ένα μοντέλο είναι πολύ απλοϊκό για να συλλάβει τα υποκείμενα μοτίβα και τις πολυπλοκότητες που υπάρχουν στα δεδομένα. Οδηγεί σε υψηλή μεροληψία (bias) και κακή απόδοση τόσο στα δεδομένα εκπαίδευσης όσο και στα δεδομένα επαλήθευσης/δοκιμής. Ένα υποπροσαρμοσμένο μοντέλο αποτυγχάνει να μάθει από τα δεδομένα εκπαίδευσης, με αποτέλεσμα την ανεπαρκή αναπαράσταση του προβλήματος και τις περιορισμένες δυνατότητες πρόβλεψης.

Από την άλλη πλευρά, η υπερπροσαρμογή συμβαίνει όταν ένα μοντέλο γίνεται υπερβολικά πολύπλοκο και συλλαμβάνει θόρυβο ή άσχετα μοτίβα στα δεδομένα εκπαίδευσης. Προσαρμόζεται πολύ στενά σρ αυτά τα δεδομένα, με αποτέλεσμα χαμηλό σφάλμα εκπαίδευσης. Ωστόσο, ένα τέτοιο μοντέλο αποδίδει ελάχιστα σε νέα, αθέατα δεδομένα, υποδεικνύοντας κακή γενίκευση και υψηλή διακύμανση (variance). Η υπερπροσαρμογή συμβαίνει συχνά όταν το μοντέλο έχει πάρα πολλές παραμέτρους ή όταν τα δεδομένα εκπαίδευσης είναι περιορισμένα.



Σχήμα 3.4: 1η εικόνα: υποπροσαρμογή, 2η εικόνα: βέλτιστη προσαρμογή, 3η εικόνα: υπερπροσαρμογή. Πηγή: [7]

Τόσο η υποπροσαρμογή όσο και η υπερπροσαρμογή είναι ανεπιθύμητες, καθώς θέτουν σε κίνδυνο την ικανότητα του μοντέλου να κάνει ακριβείς προβλέψεις σε νέα δεδομένα. Η εξισορρόπηση της πολυπλοκότητας του μοντέλου και της ποσότητας των διαθέσιμων δεδομένων εκπαίδευσης είναι υψηλής σημασίας για την εύρεση του χρυσού σημείου με-

[7] https://miro.medium.com/v2/resize:fit:750/0*hZoZam0gBfO7izIg

ταξύ υποπροσαρμογής και υπερπροσαρμογής, με αποτέλεσμα ένα μοντέλο που γενικεύει καλά και αποδίδει βέλτιστα.

Δύο όροι που αναφέραμε παραπάνω, και οι οποίοι συνδέονται σε μεγάλο βαθμό με την υποπροσαρμογή και την υπερπροσαρμογή, είναι η *διακύμανση* και η *μεροληψία* του μοντέλου (model variance, model bias).

- Η **διακύμανση μοντέλου** αναφέρεται στην ευαισθησία των προβλέψεων του μοντέλου όταν εκπαιδεύεται σε διαφορετικά υποσύνολα δεδομένων. Ένα μοντέλο με υψηλή διακύμανση είναι ιδιαίτερα ευέλικτο και ικανό να καταγράφει πολύπλοκα μοτίβα στα δεδομένα εκπαίδευσης. Ωστόσο, μπορεί επίσης να είναι υπερβολικά ευαίσθητο στο θόρυβο ή σε άσχετα χαρακτηριστικά, οδηγώντας σε υπερπροσαρμογή. Μπορεί να παρουσιάζει μεγάλες διακυμάνσεις στην απόδοση σε διαφορετικές εκτελέσεις εκπαίδευσης ή υποσύνολα δεδομένων.
- Η **μεροληψία μοντέλου** αναφέρεται στο σφάλμα ή στην απόκλιση των προβλέψεων του μοντέλου από τα πραγματικά υποκείμενα πρότυπα των δεδομένων. Αντιπροσωπεύει τις απλουστευτικές υποθέσεις ή τους περιορισμούς του μοντέλου που το κάνουν να χάνει συνεχώς σημαντικά πρότυπα ή σχέσεις. Ένα μοντέλο με υψηλή μεροληψία είναι συχνά πολύ απλοϊκό ή περιορισμένο και αποτυγχάνει να συλλάβει την πολυπλοκότητα των δεδομένων. Ένα τέτοιο μοντέλο μπορεί να μην προσαρμόζεται επαρκώς στα δεδομένα εκπαίδευσης, με αποτέλεσμα κακές επιδόσεις τόσο στο σύνολο εκπαίδευσης όσο και σε αθέατα δεδομένα. Γενικά, ένα μοντέλο με υψηλή μεροληψία παρουσιάζει συστηματικό σφάλμα ή τάση να κάνει σταθερά τους ίδιους τύπους λαθών.

Η διακύμανση και η μεροληψία του μοντέλου συζητούνται συχνά στο πλαίσιο της *αντιστάθμισης μεροληψίας-διακύμανσης*[26]. Η αντιστάθμιση αυτή υποδηλώνει ότι καθώς ένα μοντέλο γίνεται πιο πολύπλοκο και ευέλικτο (π.χ. αυξάνοντας τον αριθμό των παραμέτρων ή προσθέτοντας επίπεδα σε ένα νευρωνικό δίκτυο), η διακύμανσή του τείνει να αυξάνεται, ενώ η μεροληψία του μειώνεται. Αντίθετα, καθώς το μοντέλο γίνεται απλούστερο (π.χ. μειώνοντας τον αριθμό των παραμέτρων ή χρησιμοποιώντας ένα γραμμικό μοντέλο), η μεροληψία του αυξάνεται αλλά η διακύμανσή του μειώνεται. Ο στόχος είναι να βρεθεί μια ισορροπία μεταξύ προκατάληψης και διακύμανσης, ώστε να επιτευχθεί ένα μοντέλο που μπορεί να γενικεύσει καλά σε αόρατα δεδομένα, ενώ παράλληλα να καταγράφει τα σημαντικά μοτίβα στα δεδομένα εκπαίδευσης.

Με βάση τα παραπάνω, μπορεί κανείς εύκολα να διαπιστώσει ότι η υποπροσαρμογή συνδέεται με υψηλή μεροληψία, ενώ η υπερπροσαρμογή με υψηλή διακύμανση. Ευτυχώς, υπάρχουν διάφοροι τρόποι επίλυσης αυτών των προβλημάτων.

Αντιμετωπίζοντας το πρόβλημα της υψηλής μεροληψίας (δηλαδή της υποπροσαρμογής), η πιο απλή λύση είναι η αύξηση της πολυπλοκότητας του μοντέλου. Στο πλαίσιο των νευρωνικών δικτύων, αυτό σημαίνει αύξηση του αριθμού των κρυφών στρωμάτων ή του αριθμού των νευρώνων σε κάθε στρώμα. Μια άλλη λύση θα ήταν η αύξηση του αριθμού των δειγμάτων εκπαίδευσης.

Από την άλλη πλευρά, η αντιμετώπιση ενός μοντέλου με υψηλή διακύμανση (δηλαδή υπερπροσαρμογή) μπορεί να είναι λίγο πιο περίπλοκη. Ένας δημοφιλής τρόπος αντιμε-

τώπισης αυτού του μειονεκτήματος είναι η *κανονικοποίηση*. Η κανονικοποίηση λειτουργεί προσθέτοντας τον επιπλέον όρο $\lambda R(\mathbf{w})$ στη συνάρτηση απώλειας L, η οποία ευνοεί απλούστερα μοντέλα, με λιγότερες παραμέτρους και μικρότερες τιμές παραμέτρων. Η υπερπαράμετρος λ ελέγχει την ισχύ της κανονικοποίησης, και επιλέγεται έτσι ώστε να βελτιώνεται η απόδοση. Η συνάρτηση $R : \mathbb{R}^n \mapsto \mathbb{R}$ είναι η συνάρτηση κανονικοποίησης και ποικίλλει ανάλογα με τον τύπο της κανονικοποίησης. Οι δύο πιο συχνά χρησιμοποιούμενοι τύποι κανονικοποίησης είναι η κανονικοποίηση Lasso (*Lasso regularization*, επίσης γνωστή ως L_1 κανονικοποίηση) και η κανονικοποίηση Ridge (*Ridge regularization*, επίσης γνωστή ως L_2 κανονικοποίηση)[7]:

- Η **κανονικοποίηση Lasso** ενθαρρύνει τη σποραδικότητα στις τιμές των παραμέτρων, με αποτέλεσμα ορισμένες παράμετροι να μηδενίζονται. Τιμωρεί εξίσου τις χαμηλές και τις υψηλές τιμές παραμέτρων. Ορίζεται ως [8]:

$$R_{L_1}(\mathbf{w}) = \sum_{i=1}^{n} |w_i| \tag{3.6}$$

- Η **κανονικοποίηση Ridge** ενθαρρύνει μικρότερες τιμές παραμέτρων, τιμωρώντας σημαντικά τις υψηλές τιμές. Τείνει να κατανέμει την επίδραση κάθε χαρακτηριστικού σε όλες τις παραμέτρους, αποτρέποντας έτσι την κυριαρχία λίγων χαρακτηριστικών και βελτιώνοντας τη γενίκευση. Ορίζεται ως:

$$R_{L_2}(\mathbf{w}) = \frac{1}{2} \sum_{i=1}^{n} w_i^2 \tag{3.7}$$

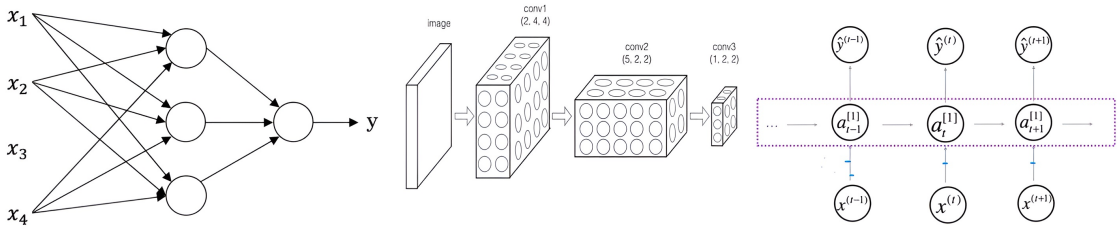
  Η επίδραση του L_2 στην ελαχιστοποίηση των τιμών των παραμέτρων είναι επίσης γνωστή ως *weight decay*.

Ένας άλλος δημοφιλής τρόπος μείωσης της υπερπροσαρμογής είναι το *dropout*[29]. Το dropout λειτουργεί αποτρέποντας το δίκτυο από το να βασίζεται υπερβολικά σε συγκεκριμένους νευρώνες ή χαρακτηριστικά. Περιλαμβάνει την τυχαία ¨απόρριψη' ενός ποσοστού των νευρώνων κατά τη διάρκεια της εκπαίδευσης, που σημαίνει ότι αγνοούνται προσωρινά ή μηδενίζονται. Αυτό ενθαρρύνει το δίκτυο να γίνει λιγότερο ευαίσθητο στην παρουσία οποιουδήποτε μεμονωμένου νευρώνα και να μάθει να βασίζεται στη συλλογική γνώση ενός μεγαλύτερου συνόλου νευρώνων.

Πιο συγκεκριμένα, κατά τη διάρκεια της εκπαίδευσης, σε κάθε επανάληψη, ένα υποσύνολο νευρώνων σε ένα στρώμα επιλέγεται τυχαία, για να ¨εγκαταλειφθεί¨, με μια συγκεκριμένη πιθανότητα (συνήθως μεταξύ 0.2 και 0.5). Αυτό σημαίνει ότι η έξοδος αυτών των νευρώνων μηδενίζεται και τα βάρη τους δεν ενημερώνονται κατά τη διάρκεια της συγκεκριμένης επανάληψης. Αυτή η διαδικασία επαναλαμβάνεται για κάθε παράδειγμα εκπαίδευσης και σε κάθε επίπεδο του δικτύου[9].

[8] Σημειώστε ότι το άθροισμα αρχίζει από το 1 και όχι από το 0. Το w_0 είναι επίσης γνωστό ως όρος προκατάληψης και δεν θεωρείται βάρος μοντέλου.

[9] Κατά τη διάρκεια της δοκιμής ή της χρήσης του δικτύου για προβλέψεις, το dropout συνήθως απενεργοποιείται και χρησιμοποιούνται όλοι οι νευρώνες. Ωστόσο, τα βάρη των νευρώνων μειώνονται κατά την πιθανότητα εγκατάλειψης, για να ληφθεί υπόψη ο μεγαλύτερος αριθμός ενεργών νευρώνων κατά την εκπαίδευση. Οι περισσότεροι σύγχρονοι αλγόριθμοι εφαρμόζουν αυτή τη διαδικασία αυτόματα.



Σχήμα 3.5: Οι πιο δημοφιλείς αρχιτεκτονικές DNN. Εικόνα 1η: το πολυεπίπεδο perceptron, Εικόνα 2η: το συνελικτικό νευρωνικό δίκτυο, Εικόνα 3η: το αναδρομικό νευρωνικό δίκτυο. Πηγή: [11]

3.4 Βαθιά μάθηση & Αρχιτεκτονικές

Τώρα που καλύψαμε τις βασικές έννοιες των νευρωνικών δικτύων, είμαστε έτοιμοι να εμβαθύνουμε σε πιο σύνθετες αρχιτεκτονικές. Το λογικό επόμενο βήμα μετά τη μελέτη του απλού νευρώνα είναι η στοίβαξη πολλαπλών νευρώνων τόσο οριζόντια όσο και κάθετα- γεννώντας έτσι μια νέα οικογένεια μοντέλων, που ονομάστηκαν εύστοχα *Βαθιά Νευρωνικά Δίκτυα* (Deep Neural Networks, DNNs).

Ένα βαθύ νευρωνικό δίκτυο είναι ένας τύπος τεχνητού νευρωνικού δικτύου που αποτελείται από πολλαπλά στρώματα διασυνδεδεμένων κόμβων ή νευρώνων. Ονομάζεται 'βαθύ' επειδή διαθέτει περισσότερα από ένα κρυφά στρώματα μεταξύ των στρωμάτων εισόδου και εξόδου, επιτρέποντάς του να μαθαίνει ιεραρχικές αναπαραστάσεις των δεδομένων εισόδου[10].

Σε αυτό το είδος δικτύου, κάθε επίπεδο αποτελείται από πολλαπλούς νευρώνες που εκτελούν υπολογισμούς στα δεδομένα εισόδου. Οι νευρώνες σε ένα στρώμα συνδέονται με τους νευρώνες στα γειτονικά στρώματα, σχηματίζοντας ένα δίκτυο διασυνδεδεμένων κόμβων. Οι συνδέσεις μεταξύ των νευρώνων συσχετίζονται με βάρη, τα οποία ρυθμίζονται ξεχωριστά κατά τη διάρκεια της εκπαίδευσης, όπως συζητήσαμε στην προηγούμενη ενότητα.

Το επίπεδο εισόδου λαμβάνει τα ακατέργαστα δεδομένα εισόδου, τα οποία μπορεί να είναι εικόνες, ήχος, κείμενο ή οποιαδήποτε άλλη μορφή δεδομένων. Τα κρυφά στρώματα, τα οποία παρεμβάλλονται μεταξύ των στρωμάτων εισόδου και εξόδου, εξάγουν σταδιακά αναπαραστάσεις υψηλότερου επιπέδου των δεδομένων εισόδου. Κάθε κρυφό στρώμα εφαρμόζει ένα σύνολο μη γραμμικών μετασχηματισμών στις εισόδους, μέσω των προαναφερθέντων συναρτήσεων ενεργοποίησης, επιτρέποντας στο δίκτυο να μαθαίνει σύνθετα μοτίβα και χαρακτηριστικά στα δεδομένα.

[10] Η ιεραρχική αναπαράσταση αναφέρεται στην ιδέα ότι κάθε επίπεδο του δικτύου μαθαίνει να εξάγει όλο και πιο αφηρημένα και σύνθετα χαρακτηριστικά από τα δεδομένα εισόδου.

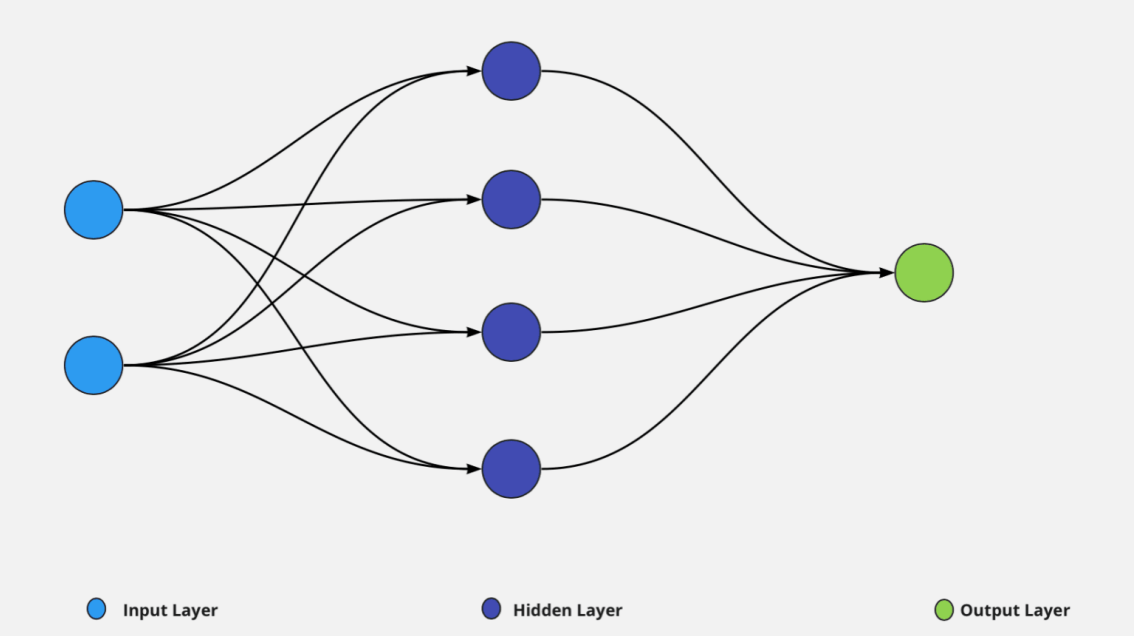
[11] https://vitalflux.com/wp-content/uploads/2021/11/deep-neural-network-examples.png

Το στρώμα εξόδου παράγει τις τελικές προβλέψεις ή εξόδους με βάση τις πληροφορίες που μαθαίνονται από τα προηγούμενα στρώματα. Ο αριθμός των νευρώνων στο στρώμα εξόδου εξαρτάται από τη συγκεκριμένη εργασία. Για παράδειγμα, σε ένα πρόβλημα ταξινόμησης, το στρώμα εξόδου μπορεί να έχει νευρώνες που αντιστοιχούν σε κάθε κλάση, ενώ σε ένα πρόβλημα παλινδρόμησης μπορεί να έχει έναν μόνο νευρώνα για την προβλεπόμενη συνεχή τιμή.

Όπως μπορεί κανείς να φανταστεί, δεν υπάρχει ένα είδος δικτύου που να ταιριάζει σε όλες τις περιπτώσεις. Κάθε κατηγορία προβλημάτων απαιτεί μια συγκεκριμένη αρχιτεκτονική, και τα προβλήματα εντός μιας δεδομένης κατηγορίας απαιτούν περαιτέρω προσαρμογές ώστε να ανταποκρίνονται στις ατομικές ανάγκες.

Παρακάτω παρουσιάζουμε τις πιο διαδεδομένες αρχιτεκτονικές, οι οποίες μπορούν να προσφέρουν λύσεις σε ένα ευρύ φάσμα εργασιών.

3.4.1 Νευρωνικά Δίκτυα Πρόσθιας Τροφοδότησης



Σχήμα 3.6: Το νευρωνικό δίκτυο πρόσθιας τροφοδότησης. Μπορούμε να παρατηρήσουμε πως τα δεδομένα ΄ρέουν΄ μόνο προς τα εμπρός. Πηγή: [12]

Τα *νευρωνικά δίκτυα πρόσθιας τροφοδότησης* (Feedforward Neural Networks, FNNs) είναι ένας τύπος τεχνητού νευρωνικού δικτύου στο οποίο η πληροφορία ρέει μόνο προς μία κατεύθυνση, από το επίπεδο εισόδου στο επίπεδο εξόδου. Ονομάζεται 'feedforward' επειδή οι συνδέσεις μεταξύ των νευρώνων δεν σχηματίζουν κύκλους ή βρόχους ανατροφοδότησης. Με άλλα λόγια, τα δεδομένα επεξεργάζονται στρώμα προς στρώμα, με την έξοδο κάθε στρώματος να χρησιμεύει ως είσοδος στο επόμενο στρώμα, χωρίς συνδέσεις ανατροφοδότησης[11]. Η απλούστερη μορφή ενός FNN, είναι το *πολυεπίπεδο Perceptron* (Multi-Layered Perceptron ,MLP).

[12] https://i0.wp.com/dataaspirant.com/wp-content/uploads/2020/09/7-feed-forward.png?resize=1536%2C1091&ssl=1

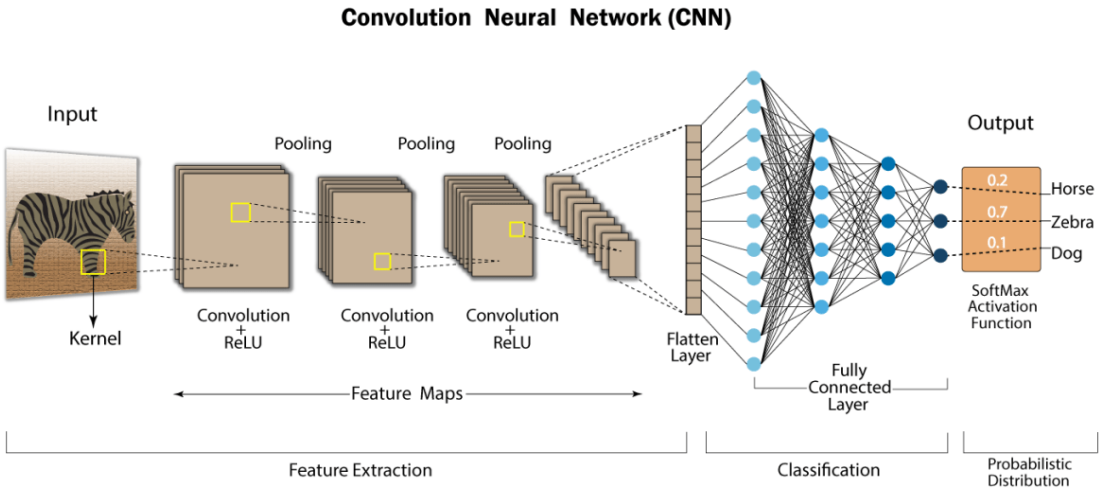
Το MLP, όπως υποδηλώνει το όνομά του, είναι ένα δίκτυο που αποτελείται από ένα επίπεδο εισόδου, ένα επίπεδο εξόδου και ένα ή περισσότερα κρυφά επίπεδα. Κάθε κρυφό στρώμα αποτελείται από πολλούς τεχνητούς νευρώνες (δηλ. perceptrons) οι οποίοι είναι πλήρως συνδεδεμένοι με τα επόμενα στρώματά τους. Κάθε στρώμα λαμβάνει είσοδο από το προηγούμενο στρώμα του, εφαρμόζει ένα σταθμισμένο άθροισμα με τα βάρη που έμαθε κατά την εκπαίδευση και περνά το αποτέλεσμα μέσω μιας μη γραμμικής συνάρτησης ενεργοποίησης. Εν τέλει, το στρώμα εξόδου αξιολογεί τις τελικές προβλέψεις.

Ο αριθμός των κρυφών στρωμάτων είναι μια ακόμη υπερπαράμετρος που πρέπει να ελεγχθεί με βάση την εκάστοτε εργασία. Ένα MLP με ένα μόνο κρυφό στρώμα είναι γνωστό ως μονοεπίπεδο perceptron (single-layer perceptron).

Τα MLP αποδίδουν καλύτερα όταν τα δεδομένα είναι σε δομημένη μορφή πίνακα, όπως αριθμητικά και κατηγορικά χαρακτηριστικά τοποθετημένα σε γραμμές και στήλες. Δυστυχώς, όμως, δεν είναι όλα τα δεδομένα σε μορφή πίνακα. Πολλές δημοφιλείς εργασίες περιλαμβάνουν εικόνες, γραφήματα, ηχητικά σήματα ή χρονοσειρές ως δεδομένα προς ανάλυση. Τα τυπικά MLP θα έπρεπε να έχουν εκατομμύρια διαφορετικά βάρη προκειμένου να επεξεργαστούν τις πληροφορίες που είναι αποθηκευμένες στα δεδομένα, ενώ παράλληλα δεν θα μπορούσαν να συλλάβουν άλλου είδους σχέσεις, όπως οι χωρικές πληροφορίες που περιέχονται στα εικονοστοιχεία (pixel) της εικόνας. Γι΄ αυτό το λόγο επεκτάθηκε η έννοια των FNNs, με νέους τύπους στρωμάτων, συνδέσεων και συνολικών αρχιτεκτονικών, τους οποίους θα εξετάσουμε.

3.4.2 Συνελικτικά Νευρωνικά Δίκτυα

Ένα *συνελικτικό νευρωνικό δίκτυο* (convolutional neural network, CNN) είναι ένα είδος FNN, σχεδιασμένο ειδικά για την επεξεργασία δομημένων δεδομένων που έχουν μορφή πλέγματος, όπως οι εικόνες.

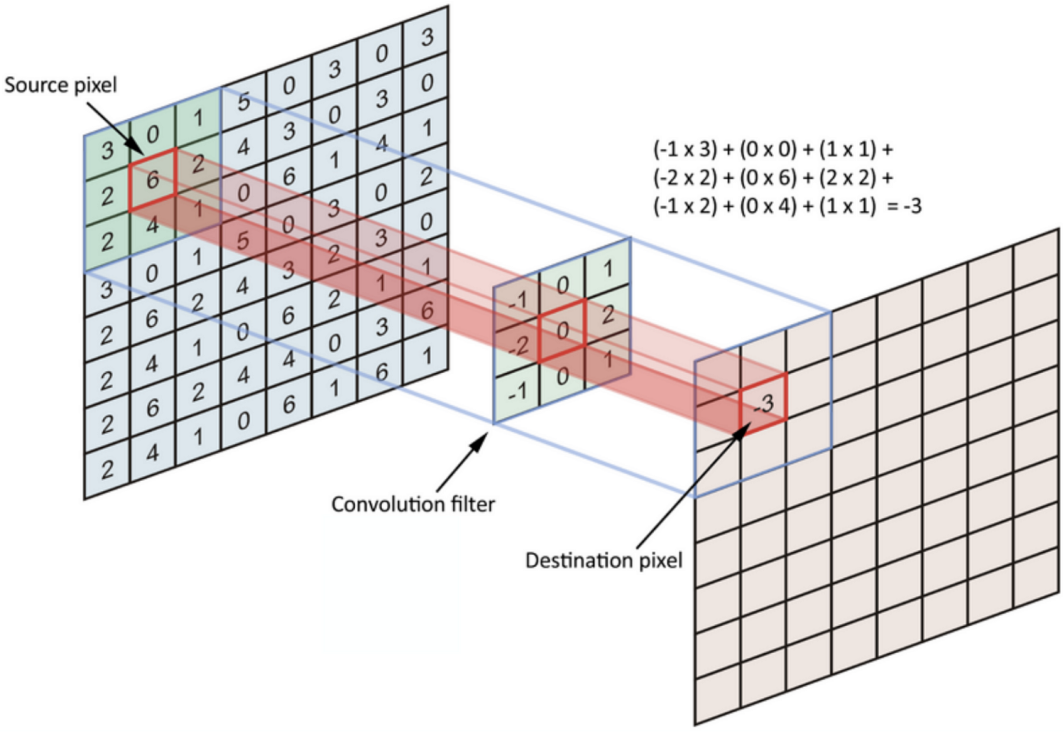


Σχήμα 3.7: Η πλήρης αρχιτεκτονική του συνελικτικού νευρωνικού δικτύου. Πηγή:[13]

[13] https://editor.analyticsvidhya.com/uploads/75059FC.png

Σε αντίθεση με τα παραδοσιακά FNNs, τα CNNs ενσωματώνουν έναν νέο τύπο στρώματος, το *συνελικτικό στρώμα* (convolutional layer). Τα συνελικτικά στρώματα εκμεταλλεύονται τις τοπικές εξαρτήσεις και τη χωρική δομή στα δεδομένα εισόδου. Λαμβάνοντας υπόψη μόνο ένα μικρό τοπικό δεκτικό πεδίο (receptive field), συλλαμβάνουν τοπικά μοτίβα και εξάγουν σχετικά χαρακτηριστικά χωρίς να επηρεάζονται από τη συνολική δομή ολόκληρης της εισόδου, ενώ παράλληλα μειώνουν την πιθανότητα υπερπροσαρμογής. Αυτό επιτυγχάνεται με τη χρήση φίλτρων.

Τα συνελικτικά φίλτρα (επίσης γνωστά ως πυρήνες) είναι μικρού μεγέθους[14], συνήθως τετραγωνικοί, πίνακες, οι οποίοι ολισθαίνουν στα δεδομένα εισόδου και μαθαίνονται κατά την εκπαίδευση. Αυτά τα φίλτρα ολισθαίνουν (ή συνελίσσονται) στα δεδομένα εισόδου με συστηματικό τρόπο. Σε κάθε θέση, το φίλτρο πολλαπλασιάζεται στοιχειωδώς με τις αντίστοιχες τιμές της εισόδου εντός του δεκτικού του πεδίου. Τα προκύπτοντα γινόμενα αθροίζονται, παράγοντας μια ενιαία τιμή που αντιπροσωπεύει την απόκριση του φίλτρου στη συγκεκριμένη περιοχή της εισόδου. Με την εφαρμογή πολλαπλών φίλτρων, δημιουργείται ένα σύνολο χαρτών χαρακτηριστικών (feature maps). Κάθε χάρτης χαρακτηριστικών αντιπροσωπεύει την απόκριση ενός συγκεκριμένου φίλτρου στην είσοδο. Αυτοί οι χάρτες αποτυπώνουν διαφορετικά τοπικά μοτίβα που υπάρχουν στα δεδομένα εισόδου.



Σχήμα 3.8: Το συνελικτικό φίλτρο. Βλέπουμε πώς το φίλτρο συλλαμβάνει τοπικές πληροφορίες για το εικονοστοιχείο-στόχο λαμβάνοντας υπόψη και τη γειτονιά του. Πάνω δεξιά είναι μια επίδειξη του τρόπου με τον οποίο χρησιμοποιείται η λειτουργία της συνέλιξης για την έξοδο ενός αριθμού ανά εικονοστοιχείο. Πηγή: [15]

Το ίδιο σύνολο φίλτρων εφαρμόζεται σε διαφορετικές περιοχές της εισόδου, γεγονός που επιτρέπει στο δίκτυο να μαθαίνει και να αναγνωρίζει παρόμοια μοτίβα σε διαφο-

[14] Τα συνήθη μεγέθη περιλαμβάνουν $3 \times 3, 5 \times 5$

[15] https://i.stack.imgur.com/YDusp.png

ρετικές χωρικές τοποθεσίες. Αυτός ο διαμοιρασμός παραμέτρων (parameter sharing) μειώνει σημαντικά τον αριθμό των μαθησιακών παραμέτρων, καθιστώντας τα CNN πιο αποδοτικά και αποτελεσματικά για το χειρισμό δεδομένων μεγάλης κλίμακας.

Οι χάρτες χαρακτηριστικών είναι επίσης ισομεταβλητοί ως προς τη μεταφορά (equivariant to translation)[78]. Αυτό σημαίνει ότι εάν τα δεδομένα εισόδου μετατοπιστούν, η έξοδος του συνελικτικού στρώματος θα μετατοπιστεί επίσης ανάλογα. Αυτή η ιδιότητα, γνωστή ως *τοπική μεταφορική αμεταβλητότητα* (local translation invariance), επιτρέπει στα συνελικτικά στρώματα να αναγνωρίζουν αποτελεσματικά μοτίβα ή χαρακτηριστικά ανεξάρτητα από τη θέση τους στην είσοδο. Αντιθέτως, τα CNNs συνήθως δεν είναι συνολικά αμετάβλητα ως προς τη μεταφορά (globally translation invariant), αλλά υπάρχουν τρόποι να εξασφαλιστεί αυτή η ιδιότητα, όπως διερευνήθηκε στα [3] και [51].

Εκτός από τα συνελικτικά στρώματα, τα CNNs περιλαμβάνουν επίσης στρώματα *συγκέντρωσης* (pooling layers) όπου υποδειγματοληπτούν τους χάρτες χαρακτηριστικών, μειώνοντας τις χωρικές διαστάσεις (πλάτος και ύψος), διατηρώντας παράλληλα τα πιο σημαντικά χαρακτηριστικά. Το στρώμα συγκέντρωσης λειτουργεί διαιρώντας τους χάρτες χαρακτηριστικών σε τοπικές περιοχές και εκτελώντας μια πράξη συνάθροισης σε κάθε περιοχή. Οι συνήθεις πράξεις συνάθροισης περιλαμβάνουν τη μέγιστη συνάθροιση (max pooling) και τη μέση συνάθροιση (average pooling). Αυτό συμβάλλει στην επίτευξη της προαναφερθείσας μεταφορικής αμεταβλητότητας, και στη μείωση της υπολογιστικής πολυπλοκότητας.

Μετά τα στρώματα συνέλιξης και συγκέντρωσης, τα CNN συχνά διαθέτουν πλήρως συνδεδεμένα στρώματα, τα οποία μοιάζουν με την παραδοσιακή αρχιτεκτονική των νευρωνικών δικτύων. Αυτά τα στρώματα λαμβάνουν τα εξαγόμενα χαρακτηριστικά και παράγουν τις τελικές προβλέψεις με βάση τις αναπαραστάσεις που έχουν μάθει. Συναρτήσεις ενεργοποίησης, όπως η ReLU, εφαρμόζονται συνήθως μετά από κάθε στρώμα για την εμφάνιση μη γραμμικότητας.

Όπως βλέπουμε, η τυπική αρχιτεκτονική των CNN αποτελείται από πολλαπλά επίπεδα. Τα αρχικά στρώματα είναι υπεύθυνα για την εξαγωγή χαρακτηριστικών χαμηλού επιπέδου, όπως η ανίχνευση ακμών, γωνιών και υφών. Καθώς τα δεδομένα ρέουν μέσω του δικτύου, τα επόμενα στρώματα μαθαίνουν πιο σύνθετα και αφηρημένα χαρακτηριστικά, συνδυάζοντας πληροφορίες από τοπικές περιοχές για να αποτυπώσουν αναπαραστάσεις υψηλότερου επιπέδου. Αυτή η ιεραρχική εξαγωγή χαρακτηριστικών επιτρέπει στα CNN να συλλαμβάνουν αποτελεσματικά τις χωρικές σχέσεις και τα μοτίβα στα δεδομένα.

3.4.3 Αναδρομικά Νευρωνικά Δίκτυα

Τα *αναδρομικά νευρωνικά δίκτυα* (Recurrent Neural Networks, RNNs) είναι μια κατηγορία νευρωνικών δικτύων, ειδικά σχεδιασμένα για να χειρίζονται σειριακά δεδομένα. Σε αντίθεση με τα νευρωνικά δίκτυα πρόσθιας τροφοδότησης, τα οποία επεξεργάζονται κάθε είσοδο ανεξάρτητα, τα RNNs διαθέτουν συνδέσεις ανατροφοδότησης που επιτρέπουν τη μεταφορά πληροφοριών από τα προηγούμενα βήματα της ακολουθίας στο τρέχον βήμα. Αυτή η ικανότητα διατήρησης και αξιοποίησης πληροφοριών από προηγούμενα βήματα καθιστά τα RNNs κατάλληλα για εργασίες που περιλαμβάνουν

σειριακά δεδομένα, όπως η επεξεργασία φυσικής γλώσσας, η αναγνώριση ομιλίας, η ανάλυση χρονοσειρών και η αναγνώριση διαδοχικής γραφής.

Το βασικό συστατικό ενός RNN είναι η αναδρομική σύνδεση, η οποία σχηματίζει έναν βρόχο μέσα στο δίκτυο, επιτρέποντας τη ροή πληροφοριών από το ένα βήμα στο επόμενο. Σε κάθε βήμα της ακολουθίας, το RNN λαμβάνει μια είσοδο και παράγει μια έξοδο, ενημερώνοντας παράλληλα την κρυφή του κατάσταση. Η κρυφή κατάσταση χρησιμεύει ως μνήμη του δικτύου, επιτρέποντάς του να καταγράφει εξαρτήσεις και μοτίβα σε διαφορετικά χρονικά βήματα.

Ο υπολογισμός σε ένα RNN εκτελείται συνήθως με τη χρήση αναδρομικών μονάδων, όπως το βασικό κελί RNN ή πιο εξελιγμένες παραλλαγές όπως τα Long Short-Term Memory (LSTM)[30] ή Gated Recurrent Unit (GRU)[10]. Αυτές οι αναδρομικές μονάδες έχουν εσωτερικά βάρη που ελέγχουν τον τρόπο με τον οποίο οι πληροφορίες από την τρέχουσα είσοδο και την προηγούμενη κρυφή κατάσταση συνδυάζονται και μεταβιβάζονται στο επόμενο βήμα.

Σε κάθε χρονικό βήμα, το βασικό κελί RNN λαμβάνει ως είσοδο ένα διάνυσμα εισόδου και την προηγούμενη κρυφή κατάσταση. Στη συνέχεια υπολογίζει μια νέα κρυφή κατάσταση συνδυάζοντας την τρέχουσα είσοδο με την προηγούμενη κρυφή κατάσταση χρησιμοποιώντας ένα σύνολο βαρών. Ο υπολογισμός εντός του βασικού κελιού RNN μπορεί να περιγραφεί από τις ακόλουθες εξισώσεις:

$$\mathbf{h}_t = \sigma(\mathbf{W}_x \mathbf{x}_t + \mathbf{W}_h \mathbf{h}_{t-1} + b) \tag{3.8}$$

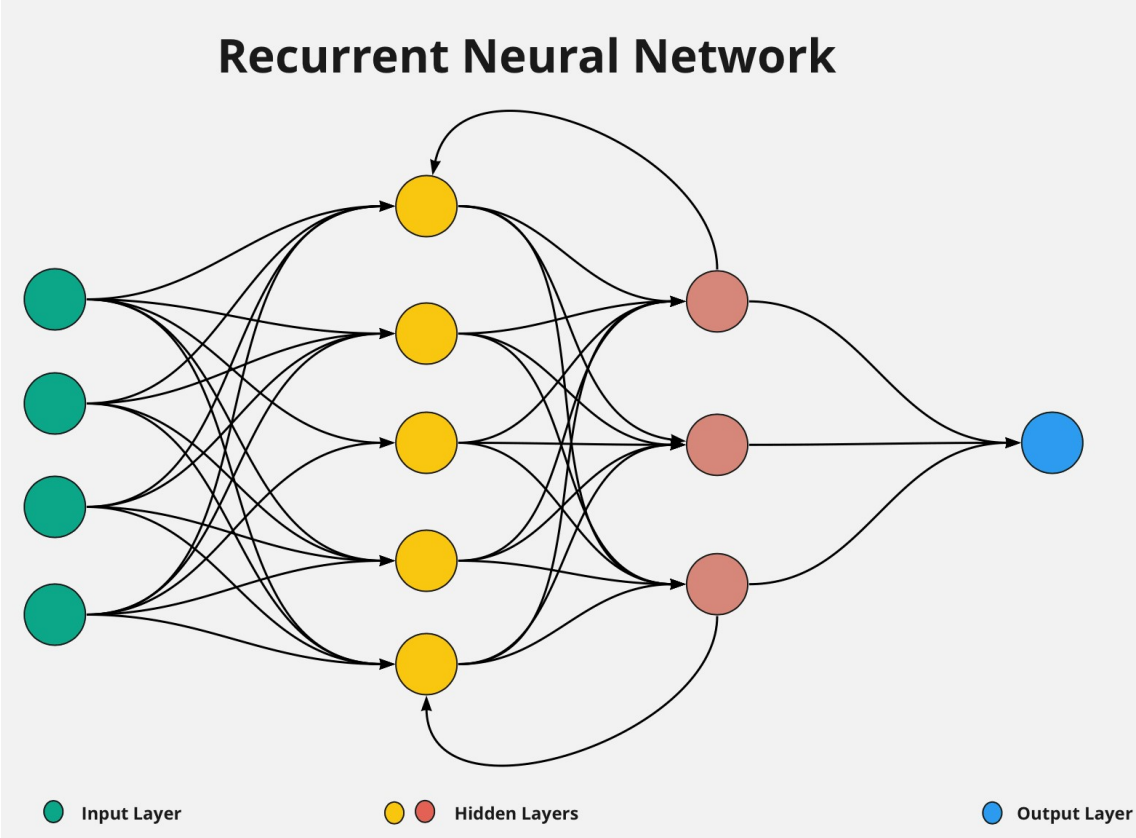
Στην παραπάνω εξίσωση:

- $\mathbf{x}_t$ αντιπροσωπεύει το διάνυσμα εισόδου στο τρέχον χρονικό βήμα,
- $\mathbf{h}_{t-1}$ αντιπροσωπεύει την κρυφή κατάσταση από το προηγούμενο χρονικό βήμα,
- $\mathbf{h}_t$ αντιπροσωπεύει τη νέα κρυφή κατάσταση στο τρέχον χρονικό βήμα
- $\mathbf{W}_x$ και $\mathbf{W}_h$ είναι πίνακες βαρών, και b είναι ο όρος bias
- σ είναι μια μη γραμμική συνάρτηση ενεργοποίησης, της επιλογής μας

Κατά τη διάρκεια της εκπαίδευσης, τα RNN μαθαίνουν να ενημερώνουν τα εσωτερικά τους βάρη χρησιμοποιώντας τον αλγόριθμο *οπισθοδιάδοσης μέσω του χρόνου* (backpropagation through time, BPTT). Ο BPTT επεκτείνει τον τυπικό αλγόριθμο οπισθοδιάδοσης ώστε να χειρίζεται τη χρονική φύση των δεδομένων και να ενημερώνει τα βάρη με βάση το σφάλμα σε όλα τα χρονικά βήματα[52][61][74].

Η αναδρομική φύση των RNNs τους επιτρέπει να μοντελοποιούν και να μαθαίνουν μοτίβα σε σειριακά δεδομένα με διαφορετικά μήκη και εξαρτήσεις με την πάροδο του χρόνου. Μπορούν να συλλάβουν πληροφορίες από το παρελθόν, γεγονός που τα καθιστά αποτελεσματικά για εργασίες όπως η πρόβλεψη ακολουθιών, η δημιουργία ακολουθιών και η διαδοχική λήψη αποφάσεων. Ωστόσο, τα RNNs μπορεί να αντιμετωπίσουν προκλήσεις με τις μακροπρόθεσμες εξαρτήσεις και μπορεί να υποφέρουν από το

πρόβλημα της εξαφανιζόμενης ή της εκρηκτικής κλίσης. Για να αντιμετωπιστούν αυτά τα ζητήματα, εισήχθησαν πιο προηγμένες παραλλαγές, όπως LSTM και GRU, οι οποίες ενσωματώνουν μηχανισμούς ελέγχου για την επιλεκτική ενημέρωση και πρόσβαση στην κρυφή κατάσταση.



Σχήμα 3.9: Το αναδρομικό νευρωνικό δίκτυο, σε απλουστευμένη μορφή. Μπορούμε να δούμε πώς οι κρυφοί νευρώνες επικοινωνούν με τα προηγούμενα επίπεδα, υλοποιώντας την ικανότητα του δικτύου να ¨απομνημονεύει¨. Πηγή: [16]

Συνοπτικά, ένα RNN είναι μια αρχιτεκτονική νευρωνικού δικτύου που έχει σχεδιαστεί για την επεξεργασία σειριακών δεδομένων διατηρώντας μνήμη μέσω αναδρομικών συνδέσεων. Αυτό επιτρέπει στο δίκτυο να μαθαίνει μοτίβα και εξαρτήσεις σε διάφορα χρονικά βήματα, καθιστώντας το κατάλληλο για εργασίες που περιλαμβάνουν ακολουθίες δεδομένων.

3.5 Διαδικασία Εκπαίδευσης Νευρωνικών Δικτύων

Έχοντας καλύψει όλα τα απαραίτητα βήματα, παρουσιάζουμε τώρα την πλήρη διαδικασία εκπαίδευσης ενός NN. Παρόλο που οι αρχιτεκτονικές και οι εργασίες μπορεί να διαφέρουν, η γενική ιδέα παραμένει η ίδια. Τα βήματα έχουν ως εξής:

[16] https://dataaspirant.com/wp-content/uploads/2020/11/3-Recurrent-Neural-Network.png

- **Προετοιμασία δεδομένων**: Προετοιμάζουμε το σύνολο δεδομένων μας, εφαρμόζοντας οποιαδήποτε βήματα προ-επεξεργασίας και χωρίζοντάς το σε σύνολα εκπαίδευσης, επαλήθευσης και δοκιμής.

- **Σχεδιασμός αρχιτεκτονικής δικτύου**: Επιλέγουμε την κατάλληλη αρχιτεκτονική για το νευρωνικό μας δίκτυο με βάση το πρόβλημα που προσπαθούμε να λύσουμε. Αυτό περιλαμβάνει την επιλογή του τύπου και του αριθμού των επιπέδων, των συναρτήσεων ενεργοποίησης και άλλων αρχιτεκτονικών στοιχείων.

- **Κάθοδος Κλίσης**: Εφαρμόζουμε τη διαδικασία καθόδου κλίσης, όπως εξηγήθηκε στην προηγούμενη ενότητα, επιλέγοντας τον κατάλληλο αλγόριθμο και επαναλαμβάνοντας τη διαδικασία για πολλές εποχές μέχρι τη σύγκλιση, εκπαιδεύοντας στο σύνολο εκπαίδευσης.

- **Επαλήθευση και ρύθμιση υπερπαραμέτρων**: Αξιολογούμε περιοδικά την απόδοση του δικτύου στο σύνολο επαλήθευσης για να παρακολουθούμε την πρόοδό του και να ρυθμίζουμε τις υπερπαραμέτρους (π.χ. ρυθμός μάθησης, ισχύς κανονικοποίησης), εάν είναι απαραίτητο.

- **Δοκιμή**: Αφού ολοκληρωθεί η εκπαίδευση, αξιολογούμε το εκπαιδευμένο δίκτυο στο σύνολο δοκιμών για να εκτιμήσουμε την απόδοσή του σε αθέατα δεδομένα.

- **Πρόβλεψη**: Αφού ολοκληρωθούν όλα τα παραπάνω βήματα, μπορούμε να χρησιμοποιήσουμε το εκπαιδευμένο μοντέλο μας για να κάνουμε νέες προβλέψεις για την εργασία μας.

Όπως αναφέρθηκε παραπάνω, ανεξάρτητα από την εργασία και την αρχιτεκτονική του μοντέλου, η εκπαίδευση ενός NN ακολουθεί συνήθως τα παραπάνω βήματα, εκτός από ακραίες περιπτώσεις όπως η διαδικτυακή ή η ενεργή μάθηση.

Έχοντας καλύψει τις βασικές έννοιες των γράφων και των νευρωνικών δικτύων, καθώς και με μια καλή κατανόηση της διαδικασίας εκπαίδευσης τέτοιων δικτύων, ήρθε η ώρα να παρουσιάσουμε το κύριο μοντέλο με το οποίο θα εργαστούμε- το Νευρωνικό Δίκτυο Γράφων.

Κεφάλαιο 4

Νευρωνικά Δίκτυα Γράφων

4.1 Εισαγωγή

Όπως διερευνήσαμε ήδη στο κεφάλαιο 2, οι γράφοι χρησιμοποιούνται ευρέως σε καθημερινές εργασίες για την αναπαράσταση σχέσεων και αλληλεπιδράσεων μεταξύ οντοτήτων σε διάφορους τομείς, όπως τα κοινωνικά δίκτυα, οι γράφοι γνώσης και τα βιολογικά δίκτυα.

Τα παραδοσιακά νευρωνικά δίκτυα είναι περιορισμένα ως προς την ικανότητά τους να χειρίζονται δεδομένα γράφων, λόγω της έλλειψης ρητής γνώσης της υποκείμενης δομής. Ενώ τα CNNs λειτουργούν καλύτερα με εικόνες, οι οποίες μπορούν να θεωρηθούν ως σταθερές δομές που μοιάζουν με πλέγμα, τα γενικά δεδομένα γράφων έχουν πολλές μορφές και μεγέθη, και τα κλασικά στρώματα συνελικτικής ανάλυσης δεν μπορούν να χρησιμοποιηθούν σωστά.

Τα νευρωνικά δίκτυα γράφων (GNN) έχουν αναδειχθεί ως ένα ισχυρό εργαλείο για την ανάλυση και την εξαγωγή πολύτιμων πληροφοριών από δεδομένα δομημένα σε μορφή γράφων. Αξιοποιώντας τα μοτίβα συνδεσιμότητας και τις σχέσεις γειτνίασης, τα GNNs μπορούν να συλλάβουν τόσο τις τοπικές όσο και τις καθολικές εξαρτήσεις, καθιστώντας τα αποτελεσματικά σε διαδικασίες όπως η ταξινόμηση κόμβων, η πρόβλεψη συνδέσμων και η εξαγωγή συμπερασμάτων σε επίπεδο γράφου.

Σε αυτό το κεφάλαιο, θα εμβαθύνουμε στα θεμέλια των GNNs, την αρχιτεκτονική τους, τη μαθηματική τους διατύπωση και τις εφαρμογές τους στον τομέα ενδιαφέροντός μας.

4.2 Τύποι Δικτύων & Εργασιών

4.2.1 Επισκόπηση εργασιών

Οι εργασίες των GNN μπορούν εύκολα να κατηγοριοποιηθούν σε τρία διαφορετικά επίπεδα, με βάση το βάθος της ανάλυσης και τη φύση του επιθυμητού αποτελέσματος:

- **Εργασίες σε επίπεδο κόμβου**: Οι εργασίες σε επίπεδο κόμβου περιλαμβάνουν την πραγματοποίηση προβλέψεων ή την εκτέλεση υπολογισμών σε επίπεδο μεμονωμένων κόμβων ενός γράφου. Αυτές οι εργασίες συνήθως αποσκοπούν στην εκμάθηση αναπαραστάσεων ή στην εξαγωγή χαρακτηριστικών για κάθε κόμβο του γράφου. Για παράδειγμα, η ταξινόμηση κόμβων είναι μια συνηθισμένη εργασία σε επίπεδο κόμβου, όπου ο στόχος είναι να αποδοθεί μια ετικέτα ή κατηγορία σε κάθε κόμβο του γράφου με βάση τα χαρακτηριστικά του ή τη δομή της γειτονιάς του. Άλλα παραδείγματα εργασιών σε επίπεδο κόμβου περιλαμβάνουν την παλινδρόμηση κόμβων, όπου ο στόχος είναι η πρόβλεψη μιας αριθμητικής τιμής για κάθε κόμβο, και την ομαδοποίηση κόμβων, όπου ο στόχος είναι η ομαδοποίηση παρόμοιων κόμβων.
- **Εργασίες επιπέδου ακμής**: Οι εργασίες σε επίπεδο ακμών επικεντρώνονται στην πραγματοποίηση προβλέψεων ή υπολογισμών με βάση τις σχέσεις μεταξύ ζευγών κόμβων (ακμών) στο γράφημα. Αυτές οι εργασίες αποσκοπούν στην

καταγραφή των αλληλεπιδράσεων ή εξαρτήσεων ανά ζεύγη μεταξύ των κόμβων. Ένα παράδειγμα μιας εργασίας σε επίπεδο ακμής είναι η πρόβλεψη συνδέσμων, όπου ο στόχος είναι να προβλεφθεί αν θα πρέπει να υπάρχει μια ακμή μεταξύ δύο κόμβων στο γράφημα. Άλλες εργασίες σε επίπεδο ακμής περιλαμβάνουν την ταξινόμηση ακμών, όπου στόχος είναι η απόδοση μιας ετικέτας σε κάθε ακμή, και την παλινδρόμηση ακμών, όπου στόχος είναι η πρόβλεψη μιας αριθμητικής τιμής που σχετίζεται με κάθε ακμή.

- **Εργασίες σε επίπεδο γραφήματος**: Οι εργασίες σε επίπεδο γραφήματος περιλαμβάνουν προβλέψεις ή υπολογισμούς που λειτουργούν σε ολόκληρο το γράφημα ως σύνολο. Αντί να εστιάζουν σε μεμονωμένους κόμβους ή ακμές, οι εργασίες αυτές εξετάζουν τη συνολική δομή και τις ιδιότητες του γράφου. Η ταξινόμηση γράφων είναι μια τυπική εργασία σε επίπεδο γράφου, όπου ο στόχος είναι η απόδοση μιας ετικέτας ή κατηγορίας σε ολόκληρο το γράφο με βάση τα δομικά χαρακτηριστικά του. Άλλα παραδείγματα εργασιών σε επίπεδο γραφήματος περιλαμβάνουν την παλινδρόμηση γραφήματος, όπου ο στόχος είναι η πρόβλεψη μιας αριθμητικής τιμής για ολόκληρο το γράφημα, και τη δημιουργία γραφήματος, όπου ο στόχος είναι η δημιουργία νέων γραφημάτων που μοιράζονται ορισμένες ιδιότητες ή χαρακτηριστικά.

4.2.2 Τύποι GNN

Εκτός από το αντικείμενο εργασίας τους, τα GNNs μπορούν επίσης να κατηγοριοποιηθούν με βάση την αρχιτεκτονική τους. Οι περισσότεροι από αυτούς τους σχεδιασμούς εμπνεύστηκαν από τους κλασικούς σχεδιασμούς των NNs και μοιράζονται κοινά στοιχεία. Οι γενικές αρχιτεκτονικές κατηγορίες είναι οι εξής:

- **Συνελικτικά Νευρωνικά Δίκτυα Γράφων (Convolutional Graph Neural Networks, CGNNs)**: Τα συνελικτικά νευρωνικά δίκτυα γράφων είναι ένας τύπος νευρωνικών δικτύων γράφων που εκτελούν λειτουργίες μεταβίβασης μηνυμάτων και συνάθροισης σε δεδομένα δομημένα σε γράφους. Έχουν χρησιμοποιηθεί ευρέως για εργασίες όπως η ταξινόμηση κόμβων, η πρόβλεψη συνδέσμων και η ταξινόμηση γράφων. Τα CGNNs συνήθως συγκεντρώνουν πληροφορίες από γειτονικούς κόμβους για να ενημερώσουν τις αναπαραστάσεις κόμβων και να συλλάβουν μοτίβα σε επίπεδο γράφου.

- **Graph Attention Networks**: Τα graph attention networks (GAT)[72] χρησιμοποιούν μηχανισμούς προσοχής (attention mechanism) για να αποδίδουν διαφορετικά βάρη σημαντικότητας σε διαφορετικούς κόμβους κατά τη διάρκεια της διαβίβασης μηνυμάτων. Αυτό επιτρέπει στο δίκτυο να εστιάζει σε πιο σημαντικούς κόμβους για συνάθροιση.

- **Αυτοκωδικοποιητές Γράφων (Graph Autoencoders)**: Οι αυτοκωδικοποιητές γράφων[37] είναι αρχιτεκτονικές νευρωνικών δικτύων που έχουν σχεδιαστεί για να μαθαίνουν αναπαραστάσεις γράφων χαμηλής διάστασης. Συνήθως αποτελούνται από έναν κωδικοποιητή που απεικονίζει το γράφημα σε έναν χώρο χαμηλότερης διάστασης και έναν αποκωδικοποιητή που ανακατασκευάζει την αρχική δομή του γράφου.

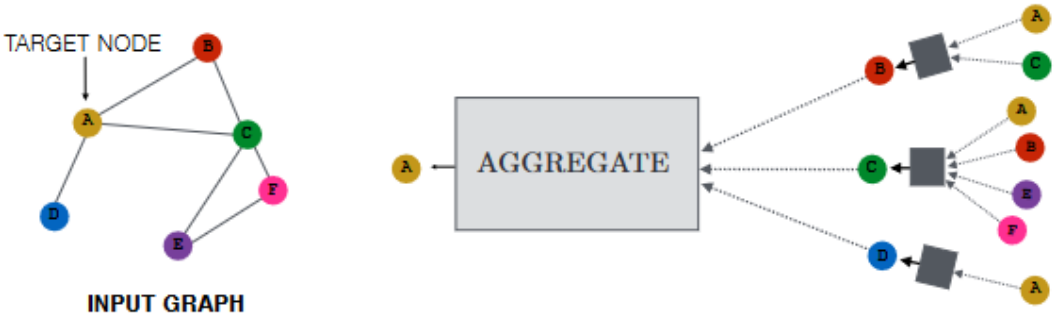
- **Μετασχηματιστές Γράφων (Graph Transformers)**: Εμπνευσμένοι από την επιτυχία των μετασχηματιστών στην επεξεργασία φυσικής γλώσσας, οι μετασχηματιστές γράφων[77] εφαρμόζουν μηχανισμούς αυτο-προσοχής (self-attention) σε γράφους για την καταγραφή παγκόσμιων εξαρτήσεων και αλληλεπιδράσεων μεταξύ κόμβων.

- **Χώρο-χρονικά Νευρωνικά Δίκτυα Γράφων (Spatial-temporal Graph Neural Networks)**: Αυτές οι αρχιτεκτονικές [63] έχουν σχεδιαστεί για να χειρίζονται δεδομένα δομημένα σε γράφους με χρονική δυναμική, όπως τα κοινωνικά δίκτυα που εξελίσσονται στο χρόνο ή τα δίκτυα κυκλοφορίας. Ενσωματώνουν τη χρονική πληροφορία για την καταγραφή δυναμικών μοτίβων και εξαρτήσεων.

4.3 Πρωτόκολλο Διαβίβασης Μηνυμάτων

Η θεμελιώδης έννοια πίσω από τα GNNs είναι η *διαβίβαση μηνυμάτων* (message-passing). Επιτρέπει την ανταλλαγή πληροφοριών μεταξύ των κόμβων ενός γράφου. Είναι ένας βασικός μηχανισμός για τη συγκέντρωση και τη διάδοση πληροφοριών σε όλη τη δομή του γράφου.

Η διαβίβαση μηνυμάτων πραγματοποιείται σε επαναλήψεις ή στρώματα, όπου κάθε κόμβος στέλνει ένα μήνυμα στους γειτονικούς του κόμβους και τα μηνύματα αυτά συγκεντρώνονται για να ενημερώσουν τις αναπαραστάσεις των κόμβων. Τα μηνύματα συνήθως καταγράφουν πληροφορίες σχετικά με την τοπική γειτονιά του κόμβου και μπορεί να περιλαμβάνουν χαρακτηριστικά, βάρη ή άλλες σχετικές πληροφορίες.



Σχήμα 4.1: Οπτική αναπαράσταση του τρόπου με τον οποίο ένας κόμβος συγκεντρώνει μηνύματα (πληροφορίες) από τους γειτονικούς του κόμβους. Ο κόμβος A συλλέγει πληροφορίες από τους κόμβους B,C,D που είχαν συλλεχθεί από τους αντίστοιχους κόμβους τους. Πρόκειται για ένα μοντέλο διακίνησης μηνυμάτων δύο επιπέδων. Πηγή: [23][Σχήμα 5.1, κεφ. 5, σελ. 49]

Πραγματοποιώντας επαναληπτική διαβίβαση μηνυμάτων, τα ΓΝΝ επιτρέπεται να αποτυπώνουν πολύπλοκα μοτίβα και εξαρτήσεις που υπάρχουν στη δομή του γράφου, ως αναπαραστάσεις κόμβων $\mathbf{h}_u$, για κάθε κόμβο $u \in V$ του γράφου.

Στην παρούσα εργασία, θα επικεντρωθούμε σε παραλλαγές των *Συνελικτικών Δικτύων Γράφων* (Graph Convolutional Networks, GCNs), οι οποίες θα εξηγηθούν λεπτομερώς

στην επόμενη ενότητα. Προς το παρόν, ορίζουμε το γενικό πρότυπο μεταβίβασης μηνυμάτων όπως παρουσιάζεται στο [9]:

$$\mathbf{h}_u = \phi\left(\mathbf{x}_u, \bigoplus_{v \in \mathcal{N}_u} c_{uv}\psi(\mathbf{x}_v)\right) \tag{4.1}$$

ωηερε:

- Οι ϕ και ψ είναι διαφορίσιμες συναρτήσεις που μαθαίνονται κατά τη διάρκεια της εκπαίδευσης, οι οποίες ονομάζονται συναρτήσεις ενημέρωσης (update) και μηνυμάτων (message), αντίστοιχα. Στο δικό μας πλαίσιο, αυτό σημαίνει νευρωνικά δίκτυα. Το πρώτο όρισμα της $\phi, \mathbf{x}_u$, αντιπροσωπεύει μια σύνδεση παράκαμψης (skip connection)[1].
- Η c_{uv} είναι μια σταθερά που καθορίζει τη σημασία του κόμβου v για την αναπαράσταση του κόμβου u. Συχνά εξαρτάται άμεσα από τα στοιχεία του πίνακα γειτνίασης $\mathbf{A}$.
- Το σύμβολο $\bigoplus$ αντιπροσωπεύει κάποιον αμετάβλητο (ή αναλλοίωτο) στις μεταθέσεις τελεστή συνάθροισης (π.χ. άθροισμα, μέσος όρος ή μέγιστο).
- Η $\mathcal{N}(u)$ είναι η γειτονιά του κόμβου u, δηλαδή $\mathcal{N}(u) = \{v \in V : (u, v) \in E\}$.

4.4 Αμεταβλητότητα και Ισοδυναμία στις μεταθέσεις

Στον ορισμό μας παραπάνω, αναφέραμε την ιδιότητα της αμεταβλητότητας στις μεταθέσεις. Μια συνάρτηση f που δέχεται ως είσοδο τον πίνακα γειτνίασης $\mathbf{A}$, ονομάζεται αμετάβλητη στις μεταθέσεις αν:

$$f(\mathbf{PAP}^T) = f(\mathbf{A}) \tag{4.2}$$

όπου P είναι ο πίνακας μετάθεσης. Στην περίπτωση που η είσοδος είναι ο πίνακας χαρακτηριστικών $\mathbf{X}$, τότε η εξίσωση 4.2 παίρνει τη μορφή:

$$f(\mathbf{PX}) = f(\mathbf{X}) \tag{4.3}$$

Γενικά, οι γράφοι μπορούν να θεωρηθούν ως μη ταξινομημένα σύνολα διανυσμάτων χαρακτηριστικών. Όταν κατασκευάζουμε τον πίνακα χαρακτηριστικών, αναπόφευκτα επιβάλλουμε μια σειρά σε αυτά τα διανύσματα χαρακτηριστικών. Αν σκοπεύουμε να χρησιμοποιήσουμε αυτά τα χαρακτηριστικά ως είσοδο για ένα NN, τότε θα θέλαμε το

[1]Μια σύνδεση παράκαμψης είναι μια σύνδεση συντόμευσης που παρακάμπτει ένα ή περισσότερα στρώματα και συνδέει απευθείας ένα προγενέστερο στρώμα με ένα μεταγενέστερο στρώμα του δικτύου.

δίκτυό μας να είναι αναλλοίωτο σε μεταθέσεις. Δηλαδή, ανεξάρτητα από τη διάταξη που επιβάλλουμε, η έξοδος δεν θα πρέπει να επηρεάζεται από αυτή τη διάταξη.

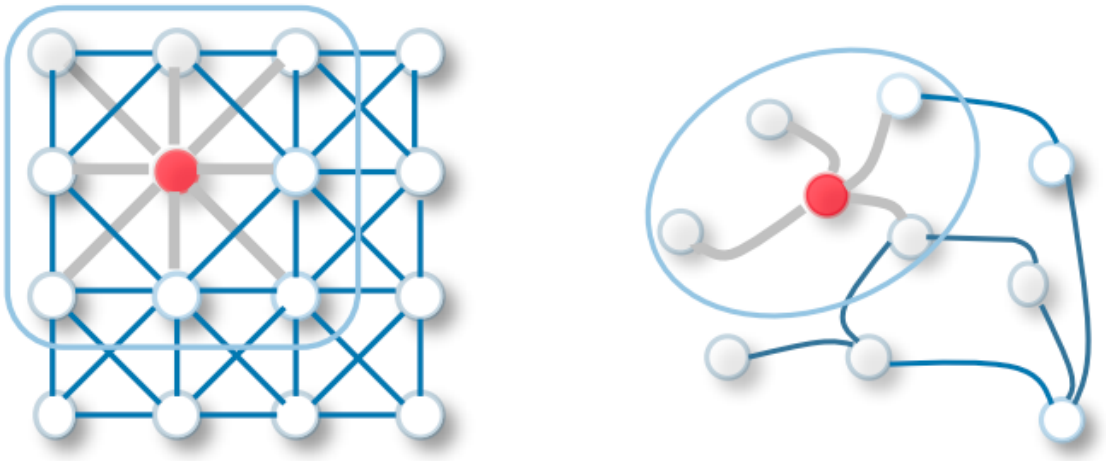
Δυστυχώς, οι αναλλοίωτες συναρτήσεις (ή τελεστές) δεν είναι τόσο συνηθισμένες. Επιπλέον, ένα δίκτυο που αποτελείται μόνο από αναλλοίωτα στρώματα δεν θα ήταν σε θέση να διακρίνει μεταξύ γραφημάτων που είναι δομικά παρόμοια αλλά έχουν διαφορετικά χαρακτηριστικά κόμβων ή συνδεσιμότητα. Όταν οι κόμβοι αντιμετωπίζονται ως σύνολα, η σειρά ή η ταυτότητα των μεμονωμένων κόμβων καθίσταται άνευ σημασίας.

Προκειμένου να βρεθεί μια ισορροπία μεταξύ αυτών των δύο προβλημάτων, συνιστάται η χρήση συναρτήσεων *ισοδύναμες στις μεταθέσεις* (ή *ευαίσθητες στις μεταθέσεις*). Για μια ισοδύναμη στις μεταθέσεις, συνάρτηση f, ισχύουν τα εξής:

$$f(\mathbf{PAP}^T) = \mathrm{P}f(\mathbf{A})\,, \text{ωηερε ινπυτ ις τηε αδθαςενςψ ματριξ } \mathbf{A}$$
$$f(\mathbf{PX}) = \mathrm{P}f(\mathbf{X})\,, \text{ωηερε ινπυτ ις τηε φεατυρε ματριξ } \mathbf{X}$$

Συνδυάζοντας αναλλοίωτα και ευαίσθητα στη μετάθεση στρώματα, τα δίκτυα γράφων μπορούν να συλλάβουν τόσο τις συνολικές ιδιότητες του γράφου όσο και τις τοπικές πληροφορίες σε επίπεδο κόμβων. Περαιτέρω ανάλυση σχετικά με τους αναλλοίωτους και ισοδύναμους τελεστές μπορεί να βρεθεί στο [45].

4.5 Συνελίξεις Γράφων



Σχήμα 4.2: Μια αναλογία μεταξύ συνελίξεων γράφων και εικόνων. Στην αριστερή εικόνα, βλέπουμε την παραδοσιακή συνέλιξη εικόνας (συνέλιξη ενός σταθερού πλέγματος), όπως παρουσιάστηκε στο κεφάλαιο 3. Στη δεξιά εικόνα, βλέπουμε τη συνέλιξη γραφήματος, ως άμεση γενίκευση των συνελίξεων στις εικόνες. Πηγή: [75][Σχήμα 1, σελ. 2]

Η συνέλιξη γράφου είναι μια πράξη που εφαρμόζει μια συνάρτηση φιλτραρίσματος ή μετασχηματισμού σε δεδομένα με δομή γράφου, όπως σήματα γράφου ή χαρακτηριστικά

που σχετίζονται με τους κόμβους ή τις ακμές ενός γράφου. Είναι εμπνευσμένη από την έννοια της συνέλιξης στην παραδοσιακή επεξεργασία σήματος, αλλά προσαρμοσμένη ώστε να λειτουργεί σε γράφους αντί για κανονικά πλέγματα (π.χ. εικόνες).

Σε μια συνέλιξη γράφου, η συνάρτηση λαμβάνει υπόψη την τοπική συνδεσιμότητα του γράφου για να αποτυπώσει τις σχέσεις μεταξύ των κόμβων. Σε αντίθεση με τη συνέλιξη σε πλέγμα, όπου ο πυρήνας συνέλιξης λειτουργεί σε γειτονιές σταθερού μεγέθους, ο πυρήνας συνέλιξης γράφου προσαρμόζεται στη δομή του γράφου, επιτρέποντας πιο ευέλικτες και εκφραστικές λειτουργίες.

Η συνέλιξη γράφου, που είναι μια μορφή μεταβίβασης μηνυμάτων, περιλαμβάνει τη συγκέντρωση πληροφοριών από γειτονικούς κόμβους ή ακμές και το συνδυασμό τους με τις πληροφορίες από τον κεντρικό κόμβο ή την κεντρική ακμή. Αυτή η συνάθροιση πραγματοποιείται συνήθως με τη χρήση ενός σταθμισμένου αθροίσματος ή μιας εκπαιδευόμενης συνάρτησης που λαμβάνει υπόψη τη συνδεσιμότητα και τη σημασία των γειτονικών στοιχείων.

Οι μέθοδοι συνέλιξης γράφων χωρίζονται σε 2 μεγάλες κατηγορίες: *Φασματικές* μέθοδοι και *χωρικές* μέθοδοι. Τα GCNs βρίσκονται ανάμεσα στη διασταύρωση αυτών των δύο κατηγοριών. Ενώ η αρχική τους διατύπωση χτίστηκε με τη χρήση ιδεών από την επεξεργασία σήματος γραφημάτων[66], βοήθησαν γρήγορα στην ανάπτυξη χωρικών μεθόδων οι οποίες προτιμώνται σε μεγάλο βαθμό λόγω του ότι κατέχουν: *΄ελκυστική αποδοτικότητα, ευελιξία και ικανότητα γενίκευσης*'[75][σελ. 7].

4.5.1 Φασματική Προσέγγιση

Οι φασματικές μέθοδοι βασίζονται στα μαθηματικά θεμέλια της θεωρίας φασμάτων γράφων και της αρμονικής ανάλυσης.[19]. Χρησιμοποιούν τη *Λαπλασιανή* του γράφου (graph Laplacian) και τη ανάλυση ιδιοτιμών της στο χώρο Fourier των συχνοτήτων, για να *φιλτράρουν* σήματα γράφων χρησιμοποιώντας τους μετασχηματισμούς Fourier γράφων.

Είναι σε θέση να συλλάβουν καθολικές πληροφορίες σχετικά με τη δομή του γράφου εξετάζοντας ολόκληρο το φάσμα του Λαπλασιανού πίνακα. Ωστόσο, τις περισσότερες φορές μας ενδιαφέρουν οι πληροφορίες που περιέχονται στην τοπική γειτονιά ενός κόμβου. Η ¨τοπικότητα' μπορεί να ενισχυθεί κάνοντας κάποιες προσεγγίσεις τοπικότητας.

Υποφέρουν επίσης από υψηλή υπολογιστική πολυπλοκότητα λόγω της εκτέλεσης ιδιοαποσύνθεσης (eigendecomposition) της Λαπλασιανής, η οποία μπορεί να είναι υπολογιστικά δαπανηρή για μεγάλους γράφους. Το πρόβλημα αυτό μπορεί επίσης να βελτιωθεί με περαιτέρω τεχνικές προσέγγισης.

4.5.2 Λαπλασιανή Γράφου

Το πρώτο βήμα για τον ορισμό της φασματικής συνέλιξης γράφων, είναι να ορίσουμε τη Λαπλασιανή γράφου.

Έστω ένας απλός μη κατευθυνόμενος γράφος $\mathcal{G} = (V, E), |V| = n$, με πίνακα γειτ-

νίασης $\mathbf{A} \in \mathbb{R}^{n \times n}$. Η βασική Λαπλασιανή γράφου (μη κανονικοποιημένη Λαπλασιανή), $\mathbf{L} \in \mathbb{R}^{n \times n}$, ορίζεται ως:

$$\mathbf{L} = \mathbf{D} - \mathbf{A}$$

όπου $\mathbf{D} \in \mathbb{R}^{n \times n}$ είναι ο διαγώνιος πίνακας βαθμών, με στοιχεία $\mathrm{D}_{ii} = \sum_j A_{ij}$.

Η Λαπλασιανή έχει κάποιες πολύ χρήσιμες ιδιότητες. Κυρίως[23][σελ. 22]:

- Είναι συμμετρική ($\mathbf{L}^T = \mathbf{L}$) και θετικά ημι-ορισμένη ($\mathbf{x}^T\mathbf{L}\mathbf{x} \geq 0, \forall \mathbf{x} \in \mathbb{R}^{|V|}$).
- Έχει $|V| = n$ στο πλήθος μη αρνητικές ιδιοτιμές. Δηλαδή: $0 = \lambda_n \leq \lambda_{n-1} \leq \cdots \leq \lambda_1$.

Η συμμετρική, κανονικοποιημένη Λαπλασιανή του $\mathcal{G}$, $\mathbf{L}^{sym} \in \mathbb{R}^{n \times n}$, ορίζεται ως:

$$\mathbf{L}^{sym} = \mathbf{I}_n - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$$

όπου:

- $\mathbf{I}_n$ είναι ο $n \times n$ μοναδιαίος πίνακας
- $\mathbf{D}^{-\frac{1}{2}} = (\mathbf{D}^{+})^{\frac{1}{2}}$, και $\mathbf{D}^{+}$ είναι ο Moore-Penrose ψευδοαντίστροφος[56].

4.5.3 Μετασχηματισμοί Φουριερ Γράφων

Το τμήμα που ακολουθεί είναι ως επί το πλείστον προσαρμοσμένο από [15] και [38].

Η βάση των GCNs χτίστηκε πάνω στη φασματική θεωρία γράφων. Ορίζουμε πρώτα ένα σήμα $\mathbf{x} : V \mapsto \mathbb{R}$ στους κόμβους του $\mathcal{G}$. Αυτό το σήμα μπορεί να θεωρηθεί ως ένα διάνυσμα $\mathbf{x} \in \mathbb{R}^n$, όπου x_i είναι η τιμή του $\mathbf{x}$ στον $i-$οστό κόμβο.

Τώρα, έστω $\mathbf{L}$ η συμμετρική κανονικοποιημένη Λαπλασιανή του $\mathcal{G}$. Αφού η $\mathbf{L}$ είναι ένας πραγματικός, συμμετρικός και θετικά ημι-ορισμένος πίνακας, μπορεί να διαγωνοποιηθεί από ένα σύνολο ορθογώνιων και κανονικοποιημένων [2] ιδιοδιανυσμάτων $\{u_k\}_{k=0}^{n-1} \in \mathbb{R}^n$, και τα αντίστοιχα διατεταγμένα, πραγματικά, μη αρνητικά ιδιοδιανύσματα $\{\lambda_k\}_{k=0}^{n-1}$. Τα ιδιοδιανύσματα ονομάζονται καταστάσεις Fourier γράφων (graph Fourier modes) και οι ιδιοτιμές ως συχνότητες γράφου. Τα ιδιοδιανύσματα δημιουργούν τη βάση Fourier $\mathbf{U} = [u_0\, u_1\, \ldots\, u_{n-1}] \in \mathbb{R}^{n \times n}$, και η διαγωνοποίηση ορίζεται από τη σχέση: $\mathbf{L} = \mathbf{U}\Lambda\mathbf{U}^T$, όπου $\Lambda = diag([\lambda_0\, \lambda_1\, \ldots\, \lambda_{n-1}]) \in \mathbb{R}^{n \times n}$.

Ο μετασχηματισμός Fourier γράφου graph Fourier transform $\hat{\mathbf{x}} \in \mathbb{R}^n$ ενός σήματος $\mathbf{x}$ ορίζεται ως:

$$\hat{\mathbf{x}} = \mathbf{U}^T\mathbf{x}$$

[2] Ορθογώνια και κανονικοποιημένα: $\langle u_i, u_j \rangle = 0, i \neq j$, και $||u_k|| = 1$, για $k = 0, 1, \ldots, n-1$, όπου $\langle \cdot, \cdot \rangle$ είναι το Ευκλείδιο εσωτερικό γινόμενο και $|| \cdot ||$ η Ευκλείδια νόρμα του $\mathbb{R}^n$

και ο αντίστροφος μετασχηματισμός ως:

$$\mathbf{x} = \mathbf{U}\hat{\mathbf{x}}$$

.

4.5.4 Φασματικές Συνελίξεις Γράφων & ChebNet

Χρησιμοποιώντας όλα τα παραπάνω, μπορούμε τώρα να ορίσουμε τις φασματικές συνελίξεις γράφων. Αυτές οι συνελίξεις μπορούν να θεωρηθούν ως ένα γινόμενο ενός σήματος $\mathbf{x}$ με ένα μη παραμετρικό φίλτρο[3] $g_\theta = diag(\theta)$. Ορίζεται από την εξίσωση:

$$g_\theta \star_\mathcal{G} \mathbf{x} = g_\theta(\mathbf{L})\mathbf{x} = g_\theta(\mathbf{U}\Lambda\mathbf{U}^T)\mathbf{x} = \mathbf{U}g_\theta(\Lambda)\mathbf{U}^T\mathbf{x}$$

Αφού το g_θ είναι μη παραμετρικό, μπορεί να θεωρηθεί ως μια συνάρτηση των ιδιοτιμών του $\mathbf{L}$, δηλαδή $g_\theta(\Lambda)$. Το g_θ παραμετροποιείται από το $\theta \in \mathbb{R}^n$, που είναι ένα διάνυσμα απο συντελεστές Fourier, και προσαρμόζεται με βάση τον δεδομένο γράφο $\mathcal{G}$ και το σήμα $\mathbf{x}$ (είναι δηλαδή μια εκπαιδεύσιμη παράμετρος).

Δυστυχώς, ο άμεσος υπολογισμός της παραπάνω συνέλιξης είναι πολύ ακριβός, καθώς περιλαμβάνει πολλαπλασιασμό πινάκων, καθώς και την ιδιοαποσύνθεση του $\mathbf{L}$, η οποία για μεγάλα γραφήματα μπορεί να είναι πρακτικά αδύνατη.Ευτυχώς, στο [25] προτάθηκε ότι το φίλτρο $g_\theta(\Lambda)$ μπορεί να προσεγγιστεί από ένα πολυωνυμικό ανάπτυγμα Chebyshev, μέχρι και K τάξη, με τη μορφή:

$$g_{\theta'}(\Lambda) = \sum_{k=0}^{K} \theta'_k T_k(\tilde{\Lambda})$$

όπου:

- $T_k(\mathrm{x})$ είναι το πολυώνυμο Chebysev, k τάξης, που ορίζεται αναδρομικά από τον τύπο: $T_k(\mathrm{x}) = 2xT_{k-1}(\mathrm{x}) - T_{k-2}(\mathrm{x}), T_0(\mathrm{x}) = 1$ και $T_1(\mathrm{x}) = \mathrm{x}$.
- $\tilde{\Lambda} = \frac{2}{\rho(\mathbf{L})}\Lambda - \mathbf{I}_n$, με $\rho(\mathbf{L})$ είναι η φασματική ακτίνα της Λαπλασιανής $\mathbf{L}$, και ορίζεται ως η μέγιστη κατά απόλυτο τιμή ιδιοτιμή του πίνακα $\mathbf{L}$, δηλαδή $\rho(\mathbf{L}) = max\{|\lambda_0|, |\lambda_1|, \ldots, |\lambda_{n-1}|\}$
- $\theta' \in \mathbb{R}^K$ είναι το ανανεωμένο διάνυσμα παραμέτρων, το οποίο αποτελείται από συντελεστές Chebyshev

Τώρα μπορούμε και πάλι να ορίσουμε τη φασματική συνέλιξη, χρησιμοποιώντας αυτή την προσέγγιση του φίλτρου, ως εξής:

[3] Ένα μη παραμετρικό φίλτρο αναφέρεται σε ένα φίλτρο που δεν έχει ένα σταθερό, προκαθορισμένο σύνολο παραμέτρων (θ). Αντίθετα, προσαρμόζεται στα χαρακτηριστικά του γραφήματος και του σήματος που επεξεργάζεται.

$$g_{\theta'} \star_{\mathcal{G}} \mathbf{x} = \sum_{k=0}^{K} \theta'_k T_k(\tilde{\mathbf{L}})\mathbf{x}$$

όπου $\tilde{\mathbf{L}} = \frac{2}{\rho(\mathbf{L})}\mathbf{L} - \mathbf{I}_n$. Η παραπάνω εξίσωση ισχύει αφού $\mathbf{L}^k = (\mathbf{U}\Lambda\mathbf{U}^T)^k = \mathbf{U}\Lambda^k\mathbf{U}^T$, και ως εκ τούτου $T_k(\tilde{\mathbf{L}}) = \mathbf{U}T_k(\tilde{\Lambda})\mathbf{U}^T$. Αυτή η συνέλιξη είναι επιπλέον K-τοπικοποιημένη, θεωρώντας ότι είναι ένα πολυώνυμο K-τάξης της Λαπλασιανής. Αυτό σημαίνει ότι λαμβάνει υπόψη τους κόμβους που βρίσκονται στη K-τάξης γειτονία από τον κεντρικό κόμβο (K βήματα μακριά). Αυτό του επιτρέπει να αποτυπώνει τοπικές πληροφορίες, ανεξάρτητα από το μέγεθος του εν λόγω γράφου, καθώς το K μπορεί να ελεγχθεί. Η παραπάνω διατύπωση είναι γνωστή ως *Chebysev Spectral CNN*, ή εν συντομία *ChebNet*.

4.5.5 Συνελικτικό Δίκτυο Γράφου

Η πρώτη προσπάθεια για τη δημιουργία ενός GCN, είναι η στοίβαξη πολλαπλών στρωμάτων συνελικτικού τύπου, το καθένα από τα οποία ακολουθείται από μια μη γραμμική συνάρτηση ενεργοποίησης (π.χ. ReLU). Όπως αναφέραμε παραπάνω, η τάξη του πολυωνύμου, K, μπορεί να επιλεγεί με βάση τη διαίσθησή μας, οδηγώντας σε διαφορετικά φίλτρα. Μια προφανής επιλογή είναι η χρήση $K = 1$, δηλαδή μια γραμμική συνάρτηση της Λαπλασιανής του γραφήματος $\mathbf{L}$. Όπως αποδεικνύεται, ένας αριθμός γραμμικών επιπέδων, στοιβαγμένων μεταξύ τους, παρέχουν επιθυμητά αποτελέσματα, ενώ παράλληλα επιλύουν το πρόβλημα της υπερπροσαρμογής, περιορίζοντας τον αριθμό των μαθησιακών παραμέτρων (θ'), το οποίο αποτελεί μια μορφή κανονικοποίησης. Επιπλέον, είναι επίσης δημοφιλής στη βιβλιογραφία η προσέγγιση $\rho(\mathbf{L}) = \lambda_{max} \approx 2$, προκειμένου να σταθεροποιηθεί η διαδικασία εκπαίδευσης και να βελτιωθεί η αριθμητική σταθερότητα. Τα πολυώνυμα ἤεβψσηε που χρησιμοποιούνται στη γραμμική προσέγγιση είναι ευαίσθητα στην κλίμακα των ιδιοτιμών και οι μεγαλύτερες ιδιοτιμές μπορεί να οδηγήσουν σε αριθμητική αστάθεια ή βραδύτερη σύγκλιση κατά τη διάρκεια της εκπαίδευσης.

Επιλέγοντας τις παραπάνω τιμές ($K = 1, \rho(\mathbf{L}) = 2$), έχουμε την προσέγγιση 1ης τάξης της διαδικασίας φιλτραρίσματος, ως:

$$g_{\theta'} \star_{\mathcal{G}} \mathbf{x} \approx \theta'_0\mathbf{x} + \theta'_1(\mathbf{L} - \mathbf{I}_n)\mathbf{x} = \theta'_0\mathbf{x} - \theta'_1\mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}\mathbf{x}$$

Όπως παρατηρήσαμε στα κανονικά CNN, τα GCN έχουν επίσης την ιδιότητα του διαμοιρασμού των βαρών (weight sharing), που σημαίνει ότι οι παράμετροι του φίλτρου θ'_0, θ'_1 είναι κοινές. Η εφαρμογή k διαδοχικών επιπέδων της παραπάνω συνέλιξης, ισοδυναμεί με τη συνέλιξη της k-τάξης γειτονιά του κεντρικού κόμβου.

Σε πρακτικές εφαρμογές, προτιμάται να κανονικοποιείται ο αριθμός των παραμέτρων ακόμη περισσότερο, ορίζοντας μια νέα παράμετρο $\theta = \theta'_0 = -\theta'_1$. Αυτό με τη σειρά του μειώνει περαιτέρω την υπερπροσαρμογή, ενώ παράλληλα μειώνει τον αριθμό των πράξεων σε κάθε στρώμα. Αυτό συνεπάγεται:

$$g_\theta \star_\mathcal{G} \mathbf{x} \approx \theta(\mathbf{I}_n + \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}})\mathbf{x}$$

Για να μειώσουμε περαιτέρω την αστάθεια (καθώς και τις εκρηκτικές\εξαφανιζόμενες κλίσεις) επανακανονικοποιούμε την έκφραση $\mathbf{I}_n + \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$ ως $\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}$, όπου $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_n$ και $\tilde{\mathrm{D}}_{ii} = \sum_j \tilde{A}_{ij}$.

Μπορούμε τώρα να παρουσιάσουμε την τελική εξίσωση. Όταν δίνεται ο πίνακας σήματος $\mathbf{X} \in \mathbb{R}^{n \times m}$ (δηλαδή πίνακας χαρακτηριστικών, όπου ένα διάνυσμα χαρακτηριστικών m-διάστασης αντιστοιχεί σε κάθε κόμβο) και F στο πλήθος φίλτρα, η γενική εξίσωση κρυφής κατάστασης για το δίκτυό μας γίνεται:

$$\mathbf{H} = g_\Theta \star_\mathcal{G} \mathbf{X} = \sigma(\bar{\mathbf{A}}\mathbf{X}\mathbf{\Theta}) \tag{4.4}$$

όπου $\mathbf{H}$ είναι ο πλήρης πίνακας αναπαράστασης των κόμβων, $\bar{\mathbf{A}} = \tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}$ και $\sigma(\cdot)$ μια κατάλληλη συνάρτηση ενεργοποίησης (ποικίλων εισόδων και εξόδων, με βάση την εργασία μας).

4.5.6 Χωρική Προσέγγιση

Τα χωρικά GCN είναι πιο στενά εμπνευσμένα από τα παραδοσιακά ΝΝ και λειτουργούν με βάση τη χωρική σχέση μεταξύ των κόμβων του γράφου. Λειτουργούν απευθείας στις τοπικές γειτονιές των κόμβων του γράφου, συγκεντρώνοντας πληροφορίες από τους άμεσους γείτονες του κεντρικού κόμβου. Εφαρμόζοντας φίλτρα που λαμβάνουν υπόψη τα χαρακτηριστικά αυτών των κόμβων και του ίδιου του κεντρικού κόμβου, ενημερώνουν την κρυφή αναπαράσταση του κεντρικού κόμβου με αναπαραστάσεις που συλλέγονται από τους γείτονες κ.ο.κ.

Τα χωρικά GCN αποτυπώνουν τοπικές πληροφορίες εντός του γράφου και είναι καταλληλότερα για το χειρισμό δομών γράφων με διαφορετικούς βαθμούς συνδεσιμότητας και ακανόνιστης μορφής, όπως τα δεδομένα μας. Έχουν χαμηλότερη υπολογιστική πολυπλοκότητα σε σύγκριση με τους φασματικούς GCN, καθώς δεν απαιτούν ιδιοαποσύνθεση, αλλά μόνο πράξεις συνάθροισης.

Όπως ήδη αναφέραμε, το GCN που ορίζεται από την εξίσωση 4.4, μπορεί να ξαναγραφεί με τη χωρική προσέγγιση ως[75]:

$$\mathbf{h}_u = \sigma\left(\mathbf{\Theta}^T\left(\sum_{v \in \mathcal{N}(u) \cup u} \bar{A}_{u,v} x_v\right)\right), \forall u \in V \tag{4.5}$$

όπου η έκφραση $\mathcal{N}(u) \cup u$ χρησιμοποιείται για να δηλώσει ότι ο κόμβος u θεωρείται γείτονας του εαυτού του.

Να σημειωθεί ότι η παραπάνω εξίσωση ακολουθεί το γενικό πρότυπο μεταβίβασης μηνυμάτων που ορίζεται στην εξίσωση 4.1.

Στις παρακάτω εξισώσεις, το $\sigma(\cdot)$ είναι μια κατάλληλη συνάρτηση ενεργοποίησης.

Ορισμένες άλλες αξιοσημείωτες χωρικές προσεγγίσεις περιλαμβάνουν:

Neural Network for Graph(NN4G)[49]: Το NN4G είναι ίσως ένα από τα πρώτα έργα που προσπάθησαν να προσεγγίσουν την έννοια των GCNs στον χώρο. Στοχεύει στην εκμάθηση των αναπαραστάσεων των κόμβων μέσω της συγκέντρωσης πληροφοριών από γειτονικούς κόμβους με τον απλούστερο τρόπο. Με τη στοίβαξη στρωμάτων συνελίξεων NN4G, μπορούμε να επεκτείνουμε τη γειτονιά των επηρεαζόμενων κόμβων. Η εξίσωση ενημέρωσης των αναπαραστάσεων για το στρώμα k και τον κόμβο u, έχει ορίζεται ως:

$$\mathbf{h}_u^{(k)} = \sigma\left(\mathbf{W}^{(k)^T}\mathbf{x}_u + \sum_{i=1}^{k-1}\sum_{v\in\mathcal{N}(u)}\mathbf{\Theta}^{(k)^T}\mathbf{h}_v^{(k-1)}\right) \tag{4.6}$$

όπου $\mathbf{W}^{(k)}, \mathbf{\Theta}^{(k)}$ είναι εκπαιδεύσιμοι πίνακες παραμέτρων, και $\mathbf{h}_u^{(0)} = 0$. Και πάλι, ακολουθεί το γενικό πρότυπο του 4.1. Μπορούμε επίσης να γράψουμε την εξίσωση 4.6 σε μορφή πινάκων:

$$\mathbf{H}^{(k)} = \sigma\left(\mathbf{X}\mathbf{W}^{(k)} + \sum_{i=1}^{k-1}\mathbf{A}\mathbf{H}^{(k-1)}\mathbf{\Theta}^{(k)}\right) \tag{4.7}$$

Μια σημαντική παρατήρηση που πρέπει να γίνει είναι ότι αυτοί οι πίνακες παραμέτρων είναι διαφορετικοί για κάθε στρώμα k. Επίσης, η εξίσωση 4.7 χρησιμοποιεί τη μη κανονικοποιημένη μορφή του πίνακα γειτνίασης $\mathbf{A}$, η οποία μπορεί να προκαλέσει σημαντικές διακυμάνσεις κλίμακας στις καταστάσεις των κόμβων $\mathbf{H}$.

Diffusion Convolutional Neural Network (DCNN)[1]: Το DCNN είναι ένας τύπος GCN που αξιοποιεί τις διαδικασίες διάχυσης για να συλλάβει πληροφορίες από τη γειτονιά των κόμβων σε έναν γράφο. Η εξίσωση για τη διαδικασία διάχυσης ορίζεται ως εξής:

$$\mathbf{H}^{(k)} = \sigma\left(\mathbf{W}^{(k)} \odot \mathbf{P}^k\mathbf{X}\right) \tag{4.8}$$

όπου $\odot$ είναι το γινόμενο πίνακα κατά στοιχείο (ή γνωστό ως γινόμενο Hadamard) και $\mathbf{P} \in \mathbb{R}^{n\times n}$ συμβολίζει τον *πίνακα πιθανοτήτων μετάβασης*, όπου ορίζεται ως $\mathbf{P} = \mathbf{D}^{-1}\mathbf{A}$. Μπορούμε επίσης να δούμε ότι ο πίνακας αναπαράστασης στο επίπεδο k, $\mathbf{H}^{(k)}$, δεν είναι συνάρτηση του $\mathbf{H}^{(k-1)}$, και το τελικό αποτέλεσμα παρουσιάζεται ως συνένωση των $\mathbf{H}^{(i)}$, για $i = 1, 2, \ldots, K$, όπου K είναι ο αριθμός των στρωμάτων (δηλαδή των επακόλουθων συνελίξεων διάχυσης).

Τα DCNN βασίζονται στην ιδέα ότι η ροή πληροφοριών στο γράφημα βασίζεται στις πιθανότητες μετάβασης που ορίζονται στο $\mathbf{P}$. Με την εφαρμογή πολλαπλών συνελίξεων, το σύστημα φτάνει σε κάποια επιθυμητή κατάσταση ισορροπίας.

4.5.7 GraphSAGE

Το μοντέλο *GraphSAGE* (*SAmple and AGgregate*) είναι μια αρχιτεκτονική GNN που έχει σχεδιαστεί για επεκτάσιμη και αποτελεσματική εκμάθηση αναπαράστασης σε γράφους μεγάλης κλίμακας. Παρουσιάστηκε από τους Hamilton et al. στο [24].

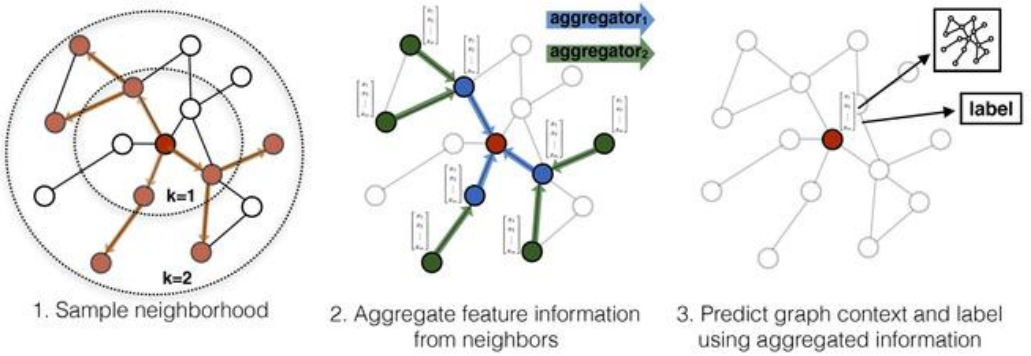


Figure 1: Visual illustration of the GraphSAGE sample and aggregate approach.

Σχήμα 4.3: Μια βήμα προς βήμα παρουσίαση του τρόπου με τον οποίο πραγματοποιείται η δειγματοληψία και η συγκέντρωση του GraphSAGE. Πηγή: [24][Σχήμα 1, σελ.2]

Οι μέχρι τώρα μέθοδοι GCN χρειάζονταν ολόκληρη την αναπαράσταση ενός γράφου για να μάθουν και δεν ήταν πολύ αποδοτικές στην κλιμάκωση. Το GraphSAGE λειτουργεί με τη δειγματοληψία της γειτονιάς ενός κόμβου και τη χρήση των χαρακτηριστικών που υπάρχουν σε αυτή τη γειτονιά για την εκπαίδευση ενός συνόλου συναρτήσεων συνάθροισης που μαθαίνουν να δημιουργούν εμφυτεύσεις κόμβων από αυτές τις πληροφορίες. Κάθε συνάρτηση συνάθροισης μαθαίνει να συλλέγει πληροφορίες μέσω διαφορετικών ακτίνων γειτονιάς, καλύπτοντας ουσιαστικά το μεγαλύτερο μέρος του γράφου. Αυτές οι συναρτήσεις μπορούν στη συνέχεια να χρησιμοποιηθούν για τη δημιουργία εμφυτεύσεων για νέους κόμβους, που δεν υπάρχουν στο σύνολο εκπαίδευσης. Αυτό καθιστά το GraphSAGE την πρώτη πραγματικά επαγωγική[4] μέθοδο, βασισμένη σε συνελίξεις.

Η γενική συνέλιξη GraphSAGE ορίζεται ως:

$$\mathbf{h}_u^{(k)} = \sigma\left(\mathbf{W}^{(k)} \cdot f_k(\mathbf{h}_u^{(k-1)}, \{\mathbf{h}_v^{(k-1)}, \forall v \in \mathcal{N}(u)\})\right) \tag{4.9}$$

όπου:

- $\mathbf{W}^{(k)}$ είναι ένας εκπαιδεύσιμος πίνακας παραμέτρων
- $\{f_k\}_{k=1}^{K}$ είναι το σύνολο των εκπαιδεύσιμων συναρτήσεων συνάθροισης
- $\mathcal{N}(u)$ είναι η γειτονιά του κόμβου u

[4]Οι μέθοδοι συνελικτικής ανάλυσης γράφων μέχρι τώρα ήταν μεταγωγικές, περιορίζονταν δηλαδή στο να κάνουν προβλέψεις μόνο για τα παρατηρούμενα σημεία δεδομένων. Οι επαγωγικές μέθοδοι μαθαίνουν να γενικεύουν και να κάνουν προβλέψεις σε νέα, αθέατα σημεία δεδομένων.

- $\mathbf{h}_u^{(0)} = \mathbf{x}_u$

Η επιλογή της συνάρτησης συνάθροισης εξαρτάται και πάλι από την εκάστοτε εργασία. Ο μόνος περιορισμός είναι ότι, γενικά, πρέπει να είναι αμετάβλητη στις μεταθέσεις. Δημοφιλείς υποψήφιες είναι οι συνολοσυναρτήσεις mean, sum ή max. Ειδικότερα, όπως περιγράφουν οι συγγραφείς στη δημοσίευσή τους, η συνάρτηση mean ως αθροιστής είναι *"σχεδόν ισοδύναμη με τον κανόνα διάδοσης της συνελικτικής διάδοσης"*[38]. Με βάση αυτό, μπορούμε να δούμε το GraphSAGE ως επαγωγική επέκταση του μεταγωγικού GCN.

Επιλέγοντας την συνάρτηση $mean(\cdot)$, παίρνουμε την απλοποιημένη μορφή[5] της σχέσης 4.9 :

$$\mathbf{h}_u^{(k)} = \sigma\left(\mathbf{W}_1\mathbf{h}_u^{(k-1)} + \mathbf{W}_2 \cdot mean(\{\mathbf{h}_v^{(k-1)}, \forall v \in \mathcal{N}(u)\})\right) \tag{4.10}$$

Άλλες, πιο σύνθετες επιλογές για συναρτήσεις του αθροιστή, που επισημαίνονται στο δημοσίευμα, είναι οι εξής:

- **Αθροιστής LSTM**: Η αρχιτεκτονική LSTM[30], ένας τύπος αναδρομικού νευρωνικού δικτύου, παρέχει μια πιο εκφραστική επιλογή συνάθροισης. Αν και εγγενώς δεν είναι αναλλοίωτη, προσαρμόζεται ώστε να λειτουργεί για μη ταξινομημένα σύνολα.
- **Αθροιστής Συγκέντρωσης (pooling aggregator)**: Ο αθροιστής συγκέντρωσης, εμπνευσμένος από το [58], χρησιμοποιεί ένα MLP για να μετασχηματίσει κάθε διανυσματική αναπαράσταση του γείτονα και στη συνέχεια συγκεντρώνει το σύνολο μετασχηματισμών που προκύπτει με τη χρήση του max(ή mean)-pooling. Αυτή η προσέγγιση μπορεί να περιγραφεί ως εξής:

  $$f_k^{pool}(\mathbf{h}_{v_i}^k) = \max_i\left(\{\mathcal{MLP}(\mathbf{h}_{v_i}^k), \forall v_i \in \mathcal{N}(u)\}\right) \tag{4.11}$$

  Στην περίπτωση ενός μονοστρωματικού perceptron, η συνάρτηση MLP γίνεται:

  $$\mathcal{MLP}(\mathbf{x}) = \sigma(\mathbf{W}_{pool}\mathbf{x} + \mathbf{b})$$

  . Ο συγκεκριμένος αθροιστής είναι και συμμετρικός.

Στη περίπτωση που αποφασίσουμε να εκπαιδεύσουμε σε minibatches, τότε αντικαθιστούμε την γειτονιά $\mathcal{N}(u)$ με ένα δείγμα της γειτονιάς, $S_{\mathcal{N}(u)}$

Το GraphSAGE κατασκευάστηκε αρχικά με στόχο την μη επιβλεπόμενη μάθηση, αλλά μπορεί επίσης να χρησιμοποιηθεί για επιβλεπόμενες ή ημι-επιβλεπόμενες διαδικασίες. Λειτουργεί εξαιρετικά καλά με γράφους που έχουν μεγάλη διακύμανση στους βαθμούς κόμβων, ανά κόμβο, όπως και το σύνολο δεδομένων μας.

[5] https://pytorch-geometric.readthedocs.io/en/latest/generated/torch_geometric.nn.conv.SAGEConv.html

Κεφάλαιο 5

Αρχεία MIDI

5.1 Εισαγωγή

Τα αρχεία MIDI (Musical Instrument Digital Interface) είναι μια ευρέως χρησιμοποιούμενη μορφή για την αναπαράσταση μουσικής σε ψηφιακή μορφή. Η μορφή ΜΙΔΙ περιέχει πληροφορίες σχετικά με τις μουσικές νότες, όπως το ύψος, τη διάρκεια και την ταχύτητά τους, καθώς και πληροφορίες σχετικά με άλλες πτυχές μιας μουσικής εκτέλεσης, όπως ο συγχρονισμός και η έκφραση. Για το λόγο αυτό, έχουν γίνει αναπόσπαστο μέρος της μουσικής παραγωγής και χρησιμοποιούνται ευρέως σε μια ποικιλία εφαρμογών, συμπεριλαμβανομένης της μουσικής σύνθεσης, της εκτέλεσης και της ηχογράφησης.

Τα τελευταία χρόνια, τα αρχεία MIDI έχουν επίσης τραβήξει προσοχή στον τομέα της μηχανικής μάθησης, ιδίως στον τομέα της δημιουργίας και επεξεργασίας μουσικής. Με την αναπαράσταση της μουσικής ως αρχεία MIDI, καθίσταται δυνατή η εφαρμογή μιας σειράς τεχνικών μηχανικής μάθησης για την ανάλυση, το μετασχηματισμό και τη δημιουργία μουσικού περιεχομένου. Αυτό έχει οδηγήσει στην ανάπτυξη μιας ποικιλίας μοντέλων και αλγορίθμων που είναι σε θέση να παράγουν νέα μουσική, να εναρμονίζουν μελωδίες, ακόμη και να αυτοσχεδιάζουν μουσικές εκτελέσεις.

Σε αυτό το κεφάλαιο, θα εξερευνήσουμε τις θεμελιώδεις έννοιες και ιδιότητες των αρχείων MIDI, συμπεριλαμβανομένης της δομής, των ιδιοτήτων και των μορφοτύπων τους. Επιπλέον, θα καλύψουμε τις βασικές ιδέες της μουσικής θεωρίας, ώστε ο αναγνώστης να εξοικειωθεί καλύτερα με τις πληροφορίες που υπάρχουν στα αρχεία MIDI.

Θα παρουσιάσουμε επίσης την κύρια ιδέα πίσω από τη δημιουργία του ετερογενούς γράφου MIDI, με βάση τις ιδέες και την υλοποίηση που συναντάμε στο [41], και πώς αυτό μπορεί να χρησιμοποιηθεί σε συνδυασμό με γνωστές αρχιτεκτονικές νευρωνικών δικτύων γράφων για την παροχή διαφόρων αποτελεσμάτων, όπως συστήματα ταξινόμησης μουσικής ή συστήματα συστάσεων.

5.2 Θεωρία Μουσικής

Δεν θα μπορούσαμε να ξεκινήσουμε τη μελέτη των αρχείων MIDI χωρίς να μιλήσουμε για τις βασικές έννοιες της μουσικής θεωρίας. Η θεωρία της μουσικής είναι η μελέτη των θεμελιωδών αρχών και στοιχείων που διέπουν τη μουσική. Παρέχει ένα πλαίσιο για την κατανόηση και την ανάλυση της δομής, της σημειογραφίας και της σύνθεσης της μουσικής. Η θεωρία της μουσικής περιλαμβάνει ένα ευρύ φάσμα θεμάτων, όπως η μελωδία, η αρμονία, ο ρυθμός, η μορφή, η δυναμική και η σημειογραφία.

Στον πυρήνα της, η θεωρία της μουσικής προσπαθεί να εξηγήσει πώς και γιατί λειτουργεί η μουσική. Εξερευνά τις σχέσεις μεταξύ των νοτών, των κλιμάκων, των συγχορδιών και των διαστημάτων και πώς δημιουργούν μια αίσθηση έντασης και λύσης. Μαθαίνοντας για αυτές τις ιδέες και τις σχέσεις, μπορούμε να κατανοήσουμε καλύτερα τις πληροφορίες που είναι αποθηκευμένες στα αρχεία MIDI και τον τρόπο με τον οποίο μπορούμε να αξιοποιήσουμε αυτές τις πληροφορίες για την εργασία μας.

Αυτή η ενότητα έχει ως στόχο να παρέχει μια σύντομη εισαγωγή στις έννοιες της

θεωρίας της μουσικής, χωρίς πολύ βαθύτερη ανάλυση. Σε περίπτωση που ο αναγνώστης επιθυμεί περαιτέρω κατανόηση του θέματος, ενθαρρύνεται να ελέγξει τα [64] ή [57].

5.2.1 Τόνος & Συχνότητα

C_3	130.81	C_4	261.63	C_5	523.25
$C^{\#}_3/D^{b}_3$	138.59	$C^{\#}_4/D^{b}_4$	277.18	$C^{\#}_5/D^{b}_5$	554.37
D_3	146.83	D_4	293.66	D_5	587.33
$D^{\#}_3/E^{b}_3$	155.56	$D^{\#}_4/E^{b}_4$	311.13	$D^{\#}_5/E^{b}_5$	622.25
E_3	164.81	E_4	329.63	E_5	659.25
F_3	174.61	F_4	349.23	F_5	698.46
$F^{\#}_3/G^{b}_3$	185.00	$F^{\#}_4/G^{b}_4$	369.99	$F^{\#}_5/G^{b}_5$	739.99
G_3	196.00	G_4	392.00	G_5	783.99
$G^{\#}_3/A^{b}_3$	207.65	$G^{\#}_4/A^{b}_4$	415.30	$G^{\#}_5/A^{b}_5$	830.61
A_3	220.00	A_4	440.00	A_5	880.00
$A^{\#}_3/B^{b}_3$	233.08	$A^{\#}_4/B^{b}_4$	466.16	$A^{\#}_5/B^{b}_5$	932.33
B_3	246.94	B_4	493.88	B_5	987.77

Σχήμα 5.1: Συχνότητες για κλίμακα ίσων τόνων, με σημείο αναφοράς $A_4 = 440Hz$, δηλαδή το τυπικό κούρδισμα[13]. Η πρώτη στήλη σε κάθε εικόνα αναπαριστά το ύψος και την οκτάβα στο τυπικό πιάνο 88 πλήκτρων (π.χ. C_4 είναι η νότα C στην οκτάβα 4^{th}). Η δεύτερη στήλη αντιπροσωπεύει τη συχνότητα της νότας (σε Hz). Πηγή: [1]

Ο *τόνος* και η *συχνότητα* είναι θεμελιώδεις έννοιες στη θεωρία της μουσικής που σχετίζονται με την αντιληπτή υψηλότητα ή χαμηλότητα ενός ήχου.

Ο τόνος αναφέρεται στην υποκειμενική αντίληψη της συχνότητας ενός ήχου. Περιγράφει τον τρόπο με τον οποίο αντιλαμβανόμαστε τους ήχους ως υψηλότερους ή χαμηλότερους με μουσικούς όρους. Ο τόνος είναι αυτό που μας επιτρέπει να διακρίνουμε τις μουσικές νότες και να αντιλαμβανόμαστε μελωδίες και αρμονίες. Στη θεωρία της μουσικής, ο τόνος αναπαρίσταται συνήθως με τα ονόματα των γραμμάτων A, B, C, κ.λπ., μαζί με σύμβολα όπως η δίεση (♯) ή η ύφεση (♭), για να υποδείξουν μεταβολές στον τόνο.

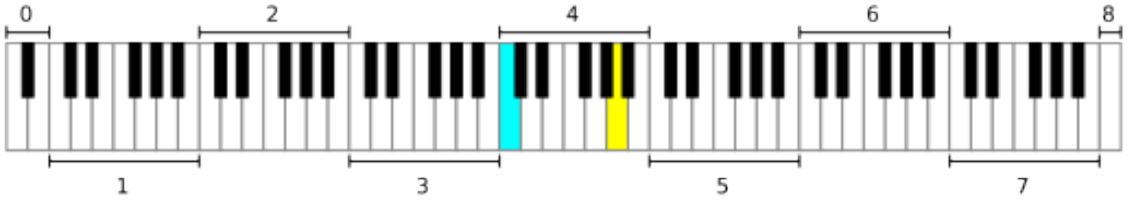
Αντιθέτως, η συχνότητα είναι μια φυσική ιδιότητα του ήχου και μετριέται σε ηερτζ (Hz). Αναφέρεται στον αριθμό των δονήσεων ή κύκλων ενός ηχητικού κύματος που συμβαίνουν σε ένα δευτερόλεπτο. Οι υψηλότερες συχνότητες αντιστοιχούν σε υψηλότερους τόνους και οι χαμηλότερες συχνότητες αντιστοιχούν σε χαμηλότερους τόνους. Για παράδειγμα, ο τόνος της μουσικής νότας A πάνω από τη μέση C είναι συνήθως συντονισμένος σε συχνότητα $440Hz$.

Ενώ ο τόνος είναι μια αντιληπτή ιδιότητα που μας επιτρέπει να ερμηνεύουμε και να κα-

[1] https://pages.mtu.edu/~suits/notefreqs.html

τηγοριοποιούμε τους ήχους, η συχνότητα είναι η αντικειμενική μέτρηση που καθορίζει τον τόνο ενός ήχου. Η κατανόηση της σχέσης μεταξύ του τόνου και της συχνότητας είναι ζωτικής σημασίας στη θεωρία της μουσικής για την κατανόηση του τρόπου με τον οποίο οι διάφορες νότες και τα διαστήματα σχετίζονται μεταξύ τους και πώς δημιουργούν τη μουσική αρμονία και τη μελωδία.

5.2.2 Νότες & Κλίμακες



Σχήμα 5.2: Το τυπικό πιάνο 88 πλήκτρων, με αριθμημένες τις οκτάβες και επισημασμένες τις μεσαίες ντο (C_4, κυανό) και A_4 (κίτρινο). Πηγή: [2]

Παρακάτω, παρουσιάζουμε ορισμένους βασικούς ορισμούς, που χρησιμοποιούνται συνήθως στη σύγχρονη θεωρία της μουσικής:

- ***Νότες***: Στο πλαίσιο της μουσικής θεωρίας, μια νότα είναι ένα σύμβολο που αντιπροσωπεύει ένα συγκεκριμένο τόνο. Κάθε νότα αντιστοιχεί σε μια συγκεκριμένη συχνότητα ή τόνο στην ακουστική κλίμακα. Οι νότες αναπαρίστανται με γράμματα από A έως G, και μπορούν να τροποποιηθούν από accidentals (π.χ. δίεση και ύφεση) για να υποδηλώσουν μεταβολές στον τόνο. Η αντιστοίχηση των γραμμάτων του λατινικού αλφαβήτου είναι η εξής: $C \rightarrow$ Ντο, $D \rightarrow$ Ρε, $E \rightarrow$ Μι, $F \rightarrow$ Φα, $G \rightarrow$ Σολ, $A \rightarrow$ Λα, $B \rightarrow$ Σι.

- ***Διαστήματα***: Ένα διάστημα αναφέρεται στην απόσταση ή το διάστημα μεταξύ δύο τόνων ή νοτών. Είναι ένα μέτρο της διαφοράς στον τόνο μεταξύ δύο μουσικών ήχων. Μετρώνται υπολογίζοντας τον αριθμό των γραμμάτων και συμπεριλαμβάνοντας τυχόν accidentals ανάμεσα στις δύο νότες. Για παράδειγμα, το διάστημα μεταξύ C και E είναι ένα τρίτο (μετρώντας τα C, D, E) και το διάστημα μεταξύ $F\sharp$ και $A\sharp$ είναι επίσης ένα τρίτο.

- ***Οκτάβα***: Η οκτάβα είναι το διάστημα ανάμεσα σε δύο νότες με το ίδιο γράμμα. Όταν μετακινούμαστε από τη μία οκτάβα στην άλλη, το ύψος της νότας είτε διπλασιάζεται (αύξουσα οκτάβα) είτε μειώνεται στο μισό (φθίνουσα οκτάβα). Για παράδειγμα, η νότα A στην οκτάβα κάτω από τη μέση C έχει τη μισή συχνότητα από τη νότα A στην οκτάβα πάνω από τη μέση C.

- ***Κλίμακα***: Μια κλίμακα είναι μια ακολουθία από νότες που παίζονται με αύξουσα ή φθίνουσα σειρά, ακολουθώντας ένα συγκεκριμένο μοτίβο διαστημάτων. Οι κλίμακες αποτελούν τη βάση για τις μελωδίες και τις αρμονίες στη μουσική. Η πιο συνηθισμένη κλίμακα είναι η μείζονα κλίμακα (major), η οποία αποτελείται

[2] By AlwaysAngry - Own work, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=20429663

από ένα συγκεκριμένο μοτίβο ολόκληρων βημάτων (W) και μισών βημάτων (H) μεταξύ των νοτών. Για παράδειγμα, το μοτίβο της μείζονος κλίμακας είναι W-W-H-W-W-W-H, που σημαίνει ότι υπάρχουν ολόκληρα βήματα μεταξύ των περισσότερων νοτών, εκτός από μισά βήματα μεταξύ της 3ης και 4ης και της 7ης και 8ης νότας.

- ***Κλειδί***: Ένα κλειδί είναι ένα συγκεκριμένο τονικό κέντρο ή μια βασική νότα γύρω από την οποία περιστρέφεται ένα μουσικό κομμάτι. Το κλειδί καθορίζει την επιλογή και τη διάταξη των νοτών και των συγχορδιών που χρησιμοποιούνται στη σύνθεση. Οι μείζονες και οι ελάσσονες κλίμακες συχνά συνδέονται με συγκεκριμένα κλειδιά, όπως το κλειδί της $C\ major$ ή το κλειδί της $A\ minor$. Η επιλογή του κλειδιού μπορεί να επηρεάσει σε μεγάλο βαθμό τη διάθεση και τον χαρακτήρα ενός μουσικού κομματιού.

- ***Τρόποι (Modes)***: Οι τρόποι είναι παραλλαγές της μείζονος και της ελάσσονος κλίμακας. Κάθε τρόπος ξεκινάει από διαφορετική νότα μέσα στην ίδια ακολουθία ολόκληρων και μισών βημάτων όπως η μείζονα ή ελάσσονα κλίμακα. Οι πιο συνηθισμένοι τρόποι περιλαμβάνουν την *Ιωνική* (μείζονα κλίμακα), την *Δοριανή*, την *Φρυγική*, την *Λυδική*, την *Μιξολυδική*, την *Αιολική* (φυσική ελάσσονα κλίμακα) και την *Λοκριανή*.

5.2.3 Ρυθμός & Μέτρο

Ο ρυθμός και το μέτρο είναι στοιχεία που παρέχουν δομή, οργάνωση και αίσθηση του χρόνου στο μουσικό κομμάτι. Ρυθμίζουν το συγχρονισμό και τη διάρκεια των μουσικών ήχων, δημιουργώντας μοτίβα και ρυθμούς που δίνουν στη μουσική τη χαρακτηριστική αίσθηση και το groove της.

Ο *ρυθμός* αναφέρεται στη διάταξη των ήχων και των σιωπών στη μουσική, δημιουργώντας μια αίσθηση κίνησης και ροής. Είναι το στοιχείο που δίνει στη μουσική τον ρυθμικό παλμό της. Τα ρυθμικά μοτίβα σχηματίζονται με την ομαδοποίηση των νοτών και των παύσεων σε συγκεκριμένες διάρκειες και ακολουθίες. Τα μοτίβα αυτά μπορεί να είναι απλά ή σύνθετα και καθορίζουν τον συνολικό ρυθμικό χαρακτήρα ενός μουσικού κομματιού.

Το *beat* (χτύπος) είναι ο υποκείμενος παλμός ή η σταθερά επαναλαμβανόμενη μονάδα χρόνου στη μουσική. Είναι η κανονική και συνεπής διαίρεση του χρόνου που καθορίζει το τέμπο ενός κομματιού. Το beat μπορεί να γίνει αισθητό ως φυσική αίσθηση ή να χτυπηθεί με το πόδι ή το χέρι. Το *μέτρο* είναι η οργάνωση των παλμών σε επαναλαμβανόμενα μοτίβα, δημιουργώντας μια αίσθηση ρυθμικής δομής. Εκφράζεται μέσω των χρονικών υπογραφών (όπως $\frac{4}{4}$, $\frac{3}{4}$, ή $\frac{6}{8}$), οι οποίες υποδεικνύουν τον αριθμό των χτύπων ανά μέτρο και το είδος της νότας που λαμβάνει έναν χτύπο. Για παράδειγμα, στο $\frac{4}{4}$, υπάρχουν 4 χτύποι σε ένα μέτρο και η νότα τετάρτου (νότα με διάρκεια ενός χτύπου) λαμβάνει έναν χτύπο. Το μέτρο καθορίζει το ρυθμικό πλαίσιο ενός μουσικού κομματιού και συμβάλλει στον καθορισμό της συνολικής αίσθησης και του χαρακτήρα του. Ένα μέτρο (επίσης γνωστό ως bar) είναι ένα τμήμα του χρόνου που περιέχει έναν συγκεκριμένο αριθμό χτύπων, όπως υποδεικνύεται από την χρονική υπογραφή (time

signature) (Το time signature στην ελληνική βιβλιογραφία συχνά συναντάται και ως μέτρο).

Οι πιο διαδεδομένες μουσικές αξίες, είναι οι εξής:

- **Ολόκληρο** (***Semibreve***): Έχει ανοιχτό οβάλ σχήμα, 𝅝, και αντιπροσωπεύει τη μεγαλύτερη διάρκεια στον κοινό χρόνο. Συνήθως διατηρείται για τέσσερις χτύπους σε $\frac{4}{4}$ χρόνο.
- **Μισό** (***Minim***): Έχει κοίλο οβάλ σχήμα με κορμό, 𝅗𝅥, και αντιπροσωπεύει το μισό της διάρκειας μιας ολόκληρης νότας. Σε $\frac{4}{4}$ χρόνο, κρατείται για δύο χτύπους.
- **Τέταρτο** (***Crotchet***): Έχει ένα γεμάτο οβάλ σχήμα με κορμό, ♩, και αντιπροσωπεύει το ένα τέταρτο της διάρκειας μιας ολόκληρης νότας. Σε $\frac{4}{4}$ χρόνο, κρατείται για ένα χτύπημα.
- **Όγδοο** (***Quaver***): Έχει οβάλ σχήμα με κορμό και σημαία, ♪. Αντιπροσωπεύει τη μισή διάρκεια μιας νότας τετάρτου. Σε χρόνο $\frac{4}{4}$, διατηρείται για μισό χτύπο.
- **Δέκατο έκτο** (***Semiquaver***): Έχει οβάλ σχήμα με κορμό και δύο σημαίες, 𝅘𝅥𝅯. Αντιπροσωπεύει τη μισή διάρκεια μιας ογδόης νότας. Σε $\frac{4}{4}$ χρόνο, κρατείται για το ένα τέταρτο του χτύπου.

Τώρα που τελειώσαμε με τους βασικούς ορισμούς της μουσικής θεωρίας, είμαστε έτοιμοι να εξερευνήσουμε τη δομή των αρχείων MIDI.

5.3 Μορφή Αρχείου MIDI

Σχήμα 5.3: Παρουσίαση των MIDI Tracks, στο πρόγραμμα FL Studio.

Η μορφή αρχείου MIDI παρέχει έναν τυποποιημένο τρόπο ανταλλαγής μουσικών δεδομένων, επιτρέποντας τη δια-λειτουργικότητα και τη συμβατότητα σε διαφορετικές πλατφόρμες και συσκευές. Περιέχουν μια ακολουθία μηνυμάτων MIDI που περιγράφουν διάφορα μουσικά γεγονότα και οδηγίες. Είναι σημαντικό να αναφέρουμε πως τα αρχεία MIDI δεν παίζουν τα ίδια μουσική (όπως π.χ. ένα αρχείο `mp3`), αλλά περιέχουν τη

μουσική πληροφορία, όπως για παράδειγμα μια παρτιτούρα. Είναι στη δικαιοδοσία του προγράμματος επεξεργασίας του αρχείου για το πως θα ερμηνευτούν αυτές οι πληροφορίες. Παρακάτω παρουσιάζουμε μια γενική επισκόπηση του τρόπου λειτουργίας των αρχείων MIDI. Οι περισσότερες πληροφορίες προέρχονται από το [50] και [70]:

- **Κεφαλίδα (Header Chunk)**: Το αρχείο MIDI ξεκινά με ένα τμήμα κεφαλίδας που παρέχει βασικές πληροφορίες, όπως ο τύπος μορφής αρχείου, ο αριθμός των κομματιών και η διαίρεση χρόνου. Αυτό το κομμάτι έχει συνήθως σταθερό μέγεθος και εμφανίζεται στην αρχή του αρχείου.
- **Track Chunks**: Μετά το κομμάτι της κεφαλίδας, ακολουθούν ένα ή περισσότερα track chunks. Κάθε track chunks αντιπροσωπεύει ένα ξεχωριστό μουσικό κομμάτι ή φωνή μέσα στο αρχείο MIDI. Τα track chunks περιέχουν μια ακολουθία γεγονότων MIDI.
- **Γεγονότα MIDI** : Στο εσωτερικό κάθε track chunk, υπάρχουν μεμονωμένα γεγονότα. Τα γεγονότα MIDI είναι τα δομικά στοιχεία των αρχείων MIDI, και αντιπροσωπεύουν διάφορες μουσικές οδηγίες και δεδομένα.

Τα γεγονότα MIDI μπορούν να κατηγοριοποιηθούν περαιτέρω σε τρεις κύριους τύπους:

- **Μετα-γεγονότα**: Τα μετα-γεγονότα (Meta events) παρέχουν μεταδεδομένα και πληροφορίες σχετικά με το αρχείο MIDI. Περιλαμβάνουν λεπτομέρειες όπως το όνομα του κομματιού, τις αλλαγές στο τεμπο, την υπογραφή του χρόνου και άλλα.
- **Γεγονότα καναλιού**: Αυτά τα γεγονότα είναι συγκεκριμένα για ένα δεδομένο κανάλι MIDI και περιλαμβάνουν μηνύματα όπως `νοτε ον/οφφ`, αλλαγές ελέγχου (control change), pitch bend και άλλες οδηγίες που αφορούν το κανάλι.
- **Αποκλειστικά γεγονότα συστήματος**: Τα αποκλειστικά γεγονότα συστήματος (System Exclusive Events, `SysEx`) μεταφέρουν δεδομένα και εντολές που αφορούν τον κατασκευαστή. Χρησιμοποιούνται για τη διαμόρφωση, τον έλεγχο και την επικοινωνία συγκεκριμένων συσκευών.

Τα συμβάντα MIDI φέρουν χρονοσήμανση για να υποδεικνύουν πότε πρέπει να αναπαραχθούν. Οι πληροφορίες χρονικής τοποθέτησης καθορίζουν το ρυθμό και το συγχρονισμό των γεγονότων MIDI. Τα αρχεία MIDI χρησιμοποιούν μια διαίρεση χρόνου για να καθορίσουν τη σχέση μεταξύ του χρόνου και των μουσικών γεγονότων. Αυτή η διαίρεση μπορεί να καθοριστεί ως παλμοί ανά τέταρτο (pulses per quarter, `PPQ`) ή καρέ ανά δευτερόλεπτο (frames per second, `FPS`) για τα SMPTE[3] χρονοκωδικοποιημένα αρχεία MIDI.

[3]Σε ένα αρχείο MIDI με βάση τον χρονοκώδικα SMPTE(*Society of Motion Picture and Television Engineers*), οι πληροφορίες χρονισμού βασίζονται στη μορφή χρονοκώδικα SMPTE και όχι στον παραδοσιακό χρονισμό MIDI. Ο χρονοκώδικας αυτός αποτελείται από ώρες, λεπτά, δευτερόλεπτα και καρέ, παρέχοντας μια ακριβή χρονική αναφορά για το συγχρονισμό. Χρησιμοποιείται στην κινηματογραφική, τηλεοπτική και οπτικοακουστική βιομηχανία για το συγχρονισμό των στοιχείων ήχου και εικόνας.

Τα αρχεία MIDI είναι ευέλικτα και μπορούν να χρησιμοποιηθούν για διάφορους σκοπούς, συμπεριλαμβανομένης της μουσικής παραγωγής, της αναπαραγωγής και του συγχρονισμού μεταξύ διαφορετικών συσκευών και λογισμικού MIDI .

5.4 Μηνύματα MIDI

Τα μηνύματα MIDI είναι μια τυποποιημένη μορφή για τη μετάδοση μουσικών πληροφοριών μεταξύ ηλεκτρονικών μουσικών συσκευών. Μεταφέρουν οδηγίες ή δεδομένα που σχετίζονται με διάφορες πτυχές της μουσικής εκτέλεσης, όπως νότες, χρονισμός, δυναμική και παραμέτρους ελέγχου. Ακολουθούν ορισμένοι συνηθισμένοι τύποι μηνυμάτων MIDI:

- `Note-On` και `Note-Off`: Αυτά τα μηνύματα υποδεικνύουν πότε μια μουσική νότα παίζεται (`Note-On`) ή αφήνεται (`Note-Off`). Περιλαμβάνουν πληροφορίες όπως ο αριθμός της νότας (τόνος), η ταχύτητα (ένταση) και το κανάλι στο οποίο μεταδίδεται το μήνυμα.
- `Control Change`: Τα μηνύματα Control Change χρησιμοποιούνται για την τροποποίηση παραμέτρων και ρυθμίσεων σε συσκευές MIDI. Μπορούν να ελέγχουν χαρακτηριστικά όπως η ένταση, το audio panning, η διαμόρφωση, το sustain και διάφορες άλλες παραμέτρους. Κάθε μήνυμα αλλαγής ελέγχου αποτελείται από έναν αριθμό ελέγχου (`CC number`) και μια τιμή.
- `Program Change`: Τα μηνύματα αλλαγής προγράμματος χρησιμοποιούνται για την επιλογή διαφορετικών ήχων οργάνων ή προεπιλογών σε μια συσκευή MIDI. Επιτρέπουν την εναλλαγή μεταξύ διαφορετικών φωνών,patches ή προγραμμάτων που ορίζονται στη συσκευή.
- `Pitch Bend`: Τα μηνύματα pitch bend ελέγχουν τη διαμόρφωση του τόνου μιας νότας. Χρησιμοποιούνται για τη δημιουργία εκφραστικών παραλλαγών του τόνου, όπως η κάμψη μιας χορδής κιθάρας ή η προσθήκη ιβρατο σε μια νότα. Τα μηνύματα pitch bend καθορίζουν το μέγεθος της απόκλισης του τόνου από το κεντρικό σημείο του τόνου.
- `System Exclusive`: Τα μηνύματα `SysEx` χρησιμοποιούνται για την επικοινωνία συγκεκριμένων συσκευών. Επιτρέπουν εκτεταμένη λειτουργικότητα και έλεγχο πέρα από τα τυπικά μηνύματα MIDI. Τα μηνύματα `ΣψσΕξ` χρησιμοποιούνται συχνά για ενημερώσεις υλικολογισμικού, ρυθμίσεις διαμόρφωσης και εξειδικευμένες εντολές που αφορούν συγκεκριμένες συσκευές MIDI.
- `Timing Clock`: Τα μηνύματα timing clock βοηθούν στο συγχρονισμό των συσκευών MIDI. Αποτελούν μέρος του πρωτοκόλλου συγχρονισμού MIDI και χρησιμοποιούνται για τη διατήρηση ακριβούς συγχρονισμού μεταξύ των συσκευών με την αποστολή παλμών ρολογιού σε τακτά χρονικά διαστήματα.
- `Polyphonic Aftertouch` και `Channel Aftertouch`: Τα μηνύματα aftertouch επιτρέπουν την εισαγωγή δεδομένων με ευαισθησία στην πίεση. Τα πολυφωνικά μηνύματα aftertouch (polyphonic aftertouch) επιτρέπουν τον ατομικό έλεγχο της

πίεσης σε κάθε παιγμένη νότα, ενώ τα μηνύματα aftertouch καναλιού (channel aftertouch) παρέχουν μια ενιαία τιμή πίεσης για όλες τις ενεργές νότες σε ένα κανάλι MIDI.

Αυτά είναι μερικά παραδείγματα μηνυμάτων MIDI που χρησιμοποιούνται συνήθως σε μουσικές παραγωγές, εκτελέσεις και εφαρμογές που βασίζονται σε MIDI.

Η διαφορά μεταξύ των γεγονότων MIDI και των μηνυμάτων είναι ότι το γεγονός αναφέρεται σε διακριτά συμβάντα που συμβαίνουν σε μια ακολουθία ή ένα chunk του MIDI και αντιπροσωπεύουν μια συγκεκριμένη μουσική ενέργεια ή εντολή. Από την άλλη πλευρά, τα μηνύματα αναφέρονται στις πραγματικές μονάδες δεδομένων που μεταφέρουν τις πληροφορίες για αυτά τα γεγονότα MIDI. Αποτελούνται από δυαδικά βψτες δεδομένων και χρησιμοποιούνται για τη μετάδοση συγκεκριμένων οδηγιών ή παραμέτρων που σχετίζονται με ένα γεγονός MIDI. Για παράδειγμα, ένα γεγονός νοτε-ον θα αναπαριστάται από ένα μήνυμα MIDI που περιλαμβάνει το στατυς βψτε που υποδεικνύει ¨νοτε-ον¨, ακολουθούμενο από τα δατα βψτες που καθορίζουν τον αριθμό της νότας, την ταχύτητα και το κανάλι.

Στην ουσία, τα γεγονότα MIDI αντιπροσωπεύουν τις μουσικές ενέργειες ή τα γεγονότα που συμβαίνουν σε μια ακολουθία MIDI, ενώ τα μηνύματα MIDI είναι οι δομές δεδομένων που κωδικοποιούν τις πληροφορίες για αυτά τα γεγονότα.

5.5 Ο τομέας του MIR & η ανάλυση αρχείων MIDI

Ο τομέας της *Ανάκτησης μουσικής πληροφορίας* (Music Information Retrieval, MIR), είναι ένας διεπιστημονικός τομέας που επικεντρώνεται στην ανάπτυξη μεθόδων και τεχνικών για την αυτόματη ανάλυση, οργάνωση και ανάκτηση πληροφοριών που σχετίζονται με τη μουσική. Συνδυάζει στοιχεία από πεδία όπως η μουσικολογία, η επεξεργασία σήματος, η μηχανική μάθηση, η επιστήμη των υπολογιστών και η ανάκτηση πληροφοριών.

Στόχος του MIR είναι η ανάπτυξη υπολογιστικών μοντέλων και αλγορίθμων που μπορούν να εξάγουν σημαντικές πληροφορίες από μουσικά σήματα ή συμβολικές αναπαραστάσεις της μουσικής (όπως αρχεία MIDI ή παρτιτούρες). Το MIR παίζει σημαντικό ρόλο σε διάφορες εφαρμογές, όπως υπηρεσίες ροής μουσικής, εξατομικευμένες λίστες αναπαραγωγής, συστήματα μουσικών συστάσεων, εργαλεία μουσικής ανάλυσης και ψηφιακές μουσικές βιβλιοθήκες. Συνδυάζει τη γνώση της μουσικής σε συγκεκριμένο τομέα με υπολογιστικές τεχνικές για να ξεκλειδώσει γνώσεις και να επιτρέψει τη βαθύτερη κατανόηση της μουσικής σε μεγάλη κλίμακα. Ο τομέας του MIR προέκυψε ως ανταπόκριση στην αυξανόμενη διαθεσιμότητα της ψηφιακής μουσικής, καθώς και στην ατομική ανάγκη για αποτελεσματική αναζήτηση, κατηγοριοποίηση και παραγωγή μουσικής και στην ανάγκη των εταιρειών να προσαρμόσουν την εμπειρία του κάθε ενός από τους χρήστες τους.

Στο πλαίσιο της ανάλυσης αρχείων MIDI, η εξαγωγή χαρακτηριστικών αναφέρεται στη διαδικασία μετατροπής των ακατέργαστων δεδομένων MIDI σε ένα σύνολο ση-

μαντικών και αντιπροσωπευτικών χαρακτηριστικών που μπορούν να χρησιμοποιηθούν ως είσοδος για αλγορίθμους μηχανικής μάθησης. Όπως είδαμε, Τα ίδια τα δεδομένα MIDI αποτελούνται από γεγονότα note on/off, velocity, duration, και control change, τα οποία δεν είναι άμεσα κατάλληλα για τους περισσότερους αλγορίθμους μηχανικής μάθησης. Η εξαγωγή χαρακτηριστικών βοηθά στη μετατροπή αυτών των ακατέργαστων δεδομένων σε μια πιο κατάλληλη αναπαράσταση που συλλαμβάνει τις σχετικές πληροφορίες για τη συγκεκριμένη ανάλυση.

Η επεξεργασία χαρακτηριστικών αναφέρεται στη διαδικασία δημιουργίας νέων χαρακτηριστικών ή μετασχηματισμού των υφιστάμενων χαρακτηριστικών σε ένα σύνολο δεδομένων για τη βελτίωση της απόδοσης των μοντέλων. Περιλαμβάνει το συνδυασμό χαρακτηριστικών που εξάγονται από το αρχείο MIDI με άλλες συμβολικές μουσικές πληροφορίες (στίχοι, πολιτιστικές πληροφορίες) για να παρέχουν χρήσιμα δεδομένα στον αλγόριθμο.

Η ανάλυση MIDI αποσκοπεί στην εξαγωγή σχετικών πληροφοριών από τα αρχεία MIDI για την επίτευξη ενός από τους προαναφερθέντες στόχους του MIR. Η συνήθης πρακτική στον τομέα αυτό είναι η χρήση μεθόδων εξαγωγής χαρακτηριστικών και μηχανικής χαρακτηριστικών προκειμένου να ¨τροφοδοτηθούν' κλασικά μοντέλα μη βαθιάς μάθησης και να παραχθεί η επιθυμητή έξοδος. Δυστυχώς, η επεξεργασία χαρακτηριστικών βασίζεται στην ανθρώπινη διαίσθηση και, ως εκ τούτου, μπορεί να παραβλέψει κρυφούς συσχετισμούς, ενώ παράλληλα πάσχει σε επεκτασιμότητα και απόδοση όταν πρόκειται για μεγάλα σύνολα δεδομένων και πολλαπλές κλάσεις[47]. Περισσότερες μελέτες για εργασίες που περιλαμβάνουν συμβολικά μουσικά δεδομένα μπορείτε να βρείτε στο [14].

Μια άλλη προσέγγιση είναι η χρήση των λεγόμενων διανυσματικών προσεγγίσεων, οι οποίες υπολογίζουν διανυσματικές εμφυτεύσεις των ηχητικών σημάτων και στη συνέχεια χρησιμοποιούν μοντέλα βαθιάς μάθησης για την εκτέλεση εργασιών ταξινόμησης. Πρόκειται για έναν αναπτυσσόμενο ακόμη τομέα και δεν έχει αντιμετωπίσει την ταξινόμηση χαρακτηριστικών υψηλού επιπέδου, όπως το είδος ή το κίνημα.

Αυτή η εργασία είναι μια πρώιμη επέκταση ήδη υπάρχουσων ιδεών στον τομέα της ανάλυσης με χρήση βαθιών νευρωνικών δικτύων, με στόχο την ανακάλυψη νέων τεχνικών πλαισίων που μπορούν να εφαρμοστούν στην κατηγοριοποίηση μουσικών αρχείων και σε συστήματα συστάσεων.

Κεφάλαιο 6

Σχεδιασμός πειράματος: Προτεινόμενος Γράφος και Μοντέλο

6.1 Αναπαράσταση με Ετερογενή Γράφο

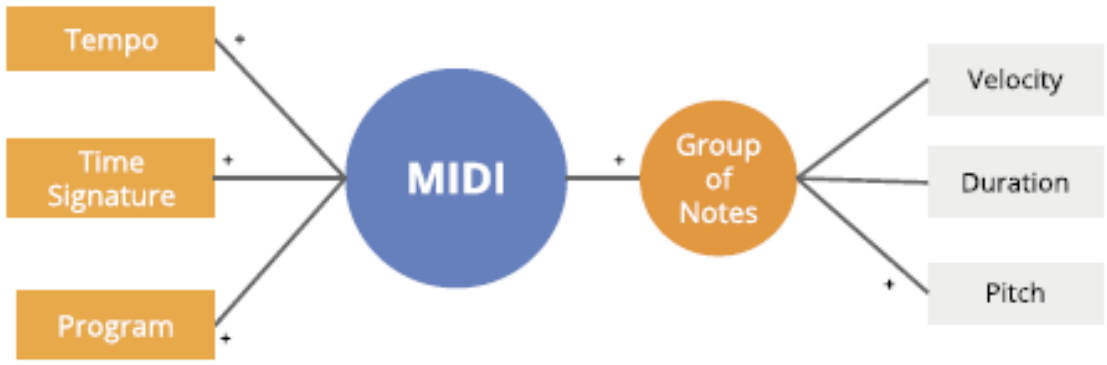
Σε αυτό το μέρος, θα παρουσιάσουμε τον τρόπο με τον οποίο κατασκευάσαμε τον ετερογενή γράφο MIDI.

6.1.1 MIDI σε Γράφο

Το πρώτο βήμα στην κατασκευή χρησιμοποιεί την υλοποίηση που δημιουργήθηκε από τους [41]. Όλες οι πληροφορίες που ακολουθούν, προέρχονται απευθείας από την προαναφερθείσα δημοσίευση.

Στην δημοσίευση, οι συγγραφείς προτείνουν μια μεθοδολογία όπου πρώτα κατασκευάζουν έναν γράφο από τις πληροφορίες που περιέχονται στο σύνολο δεδομένων MIDI (που ονομάζεται `midi2edglist`) και στη συνέχεια παράγουν μια εμφύτευση του γράφου χρησιμοποιώντας τον αλγόριθμο `node2vec`.[1]. Κρατάμε μόνο το πρώτο μέρος αυτής της διαδικασίας και το χρησιμοποιούμε στην πρωτότυπη μορφή του. Δεν έχουμε κάνει καμία αλλαγή στον πρώτο αλγόριθμο.

Ο αρχικός γράφος που κατασκευάζεται από τον αλγόριθμο `midi2edglist` αναπαριστάται στην μνήμη ως ένας ομογενής γράφος. Όμως, όπως θα δούμε παρακάτω, από τον ορισμό του, έχει σχεδιαστεί να είναι ένας ετερογενής γράφος με διαφορετικές κατηγορίες κόμβων και ακμών. Θα προσπαθήσουμε να εμπλουτίσουμε αυτές τις κατηγορίες και να δώσουμε τη δυνατότητα στο μοντέλο μας να χρησιμοποιήσει αυτές τις κατηγορίες για εξαγωγή πληροφοριών σχετικές με το είδος κάθε κόμβου.



Σχήμα 6.1: Το σχήμα γράφου MIDI. Βλέπουμε τις διάφορες κατηγορίες κόμβων: MIDI M (μπλε κόμβοι), Content C (πορτοκαλί κόμβοι) με την περαιτέρω διάκριση ότι η υποκατηγορία Note N έχει στρογγυλό σχήμα, και Attributes A (γκρι κόμβοι). Το σύμβολο + υποδεικνύει συνδέσεις ακμών τύπου πολλά προς πολλά. Πηγή: [41][Σχήμα 1, σελ. 361]

Στη συνέχεια, θα μιλήσουμε για τις επιμέρους κατηγορίες και τον τρόπο με τον οποίο οι πληροφορίες τους αναπαρίστανται στα χαρακτηριστικά των κόμβων:

[1]Και οι δύο αυτές υλοποιήσεις μπορούν να βρεθούν στο https://github.com/midi-ld/midi2vec.[20]

- **Tempo**: Το tempo θεωρείται συνεχής τιμή, οπότε πρέπει να τη χωρίσουμε σε ομάδες. Το tempo του MIDI, Tempo_{midi}, μετατρέπεται από microseconds per beat σε beats per minute (`bpm`), σύμφωνα με τον τύπο:

$$\text{Tempo}_{bpm} = \frac{60 \times 10^6}{\text{Tempo}_{midi}} \tag{6.1}$$

  Στη συνέχεια, χωρίζονται σε ομάδες των 10 `bpm`, π.χ. το `tempo-10` περιέχει το εύρος τιμών 100 ± 5`bpm`, το `tempo-6` το εύρος τιμών 60 ± 5`bpm`, κλπ.

- **Program**: Ένα από τα 128 διαφορετικά τυποποιημένα προγράμματα, που αντιπροσωπεύει το ηχόχρωμα των καναλιών. Π.χ. `program-0` είναι το ακουστικό πιάνο Acoustic Grand Piano. Ο πλήρης κατάλογος των τυποποιημένων προγραμμάτων MIDI μπορεί να βρεθεί στη διεύθυνση https://jazz-soft.net/demo/GeneralMidi.html.

- **Time signature**: Αντιπροσωπεύει το μέτρο του MIDI. Π.χ. `ts-4/4` αντιπροσωπεύει $\frac{4}{4}$.

- **Group Of Notes**: Αντιπροσωπεύει ταυτόχρονες νότες, παιγμένες μαζί. Το αρχείο MIDI δεν αναπαριστά άμεσα πληροφορίες σχετικά με τη διάρκεια και τη συνύπαρξη των νοτών (π.χ. συγχορδίες που σχηματίζονται από πολλαπλές νότες). Η εξαγωγή της διάρκειας περιλαμβάνει τη σύγκριση διαδοχικών συμβάντων `Note-On` και `Note-Off`, που βρίσκονται στο ίδιο κανάλι και έχουν τον ίδιο τόνο. Η ανίχνευση συνυπάρχουσων νοτών απαιτεί τη σύγκριση γεγονότων της ίδιας κατηγορίας σε όλα τα κανάλια και την επιλογή εκείνων με επικαλυπτόμενους δείκτες θέσης τραγουδιού (Song Position Pointers, `SPP`). Για να ενσωματώσουμε αυτές τις πληροφορίες στο γράφημα, διατηρώντας παράλληλα τον αριθμό των κόμβων και των ακμών διαχειρίσιμο, εξάγουμε ομάδες από νότες που ξεκινούν από το ίδιο `SPP`, οι οποίες υποδεικνύονται από ένα μήνυμα `Note-On`. Για να ληφθούν υπόψη πιθανές μικρές διαφορές που προκύπτουν από την καταγραφή, επιτρέπουμε μια ανοχή 10 `ms` κατά την εξέταση ταυτόχρονων νοτών. Κάθε επιμέρους ομάδα νοτών συνδέεται με 3 διαφορετικές κατηγορίες κόμβων, που περιέχουν διαφορετικές πληροφορίες για τη συγκεκριμένη ομάδα. Οι κατηγορίες αυτές είναι:

  - **Velocity**: Η μέση ταχύτητα της ομάδας. Η ταχύτητα της νότας αναφέρεται στην ένταση ή τη δύναμη με την οποία παίζεται μια συγκεκριμένη νότα. Αντιπροσωπεύει την ταχύτητα ή τη δύναμη με την οποία πατιέται ένα πλήκτρο σε ένα πληκτρολόγιο MIDI ή σε μια άλλη συσκευή εισόδου MIDI. Η ταχύτητα τόνου μετράται συνήθως σε μια κλίμακα από το 0 έως το 127, όπου το 0 υποδηλώνει καμία ταχύτητα (κανένα πλήκτρο δεν έχει πατηθεί) και το 127 αντιπροσωπεύει τη μέγιστη ταχύτητα (μέγιστη δύναμη που εφαρμόζεται στο πλήκτρο). Π.χ. `velocity-0`, `velocity-127`.
  - **Duration**: Η μέγιστη διάρκεια των νοτών που υπάρχουν στην ομάδα. Χωρίζονται σε κλάσεις των 100 `ms`. Π.χ. `duration-1` αντιπροσωπεύει το εύρος 100 ± 50 `ms`.
  - **Pitch**: Οι τόνοι που περιλαμβάνονται στην ομάδα, σύμφωνα με τους τυποποιημένους αριθμούς MIDI. Π.χ. `note-69` είναι A_4. Η συχνότητα f μπορεί

να υπολογιστεί από τον αριθμό MIDI n, σε τυπικό ισοδύναμο κούρδισμα ($A_4 = 440Hz$), από τον τύπο[2]:

$$f = 440 \cdot 2^{\frac{n-69}{12}} \tag{6.2}$$

Σε κάθε ομάδα εκχωρείται ένα αναγνωριστικό που παράγεται ντετερμινιστικά με την εφαρμογή μιας hash function στο περιεχόμενό της, και οποιεσδήποτε δύο πανομοιότυπες ομάδες θα έχουν το ίδιο αναγνωριστικό.

Όπως βλέπουμε στο Σχήμα 6.1, οι περισσότεροι κόμβοι στο γράφημά μας έχουν συνδέσεις τύπου many-to-many (υποδεικνύεται από το σύμβολο +). Αυτό σημαίνει ότι, ένας κόμβος MIDI μπορεί να συνδέεται με διαφορετικά τέμπο ή μέτρα, υποδηλώνοντας μια αλλαγή τέμπο ή μέτρων στο κομμάτι, αντίστοιχα. Επιπλέον, ένα MIDI μπορεί να συνδεθεί με διαφορετική ομάδα νοτών, καθώς και μια ομάδα νοτών μπορεί να συνδεθεί με διαφορετικούς κόμβους MIDI. Μια ομάδα νοτών μπορεί όμως να συνδέεται μόνο με έναν κόμβο velocity και duration, ενώ ο ίδιος κόμβος duration (ή velocity) μπορεί να συνδέεται με πολλές διαφορετικές ομάδες νοτών.

Οι συγγραφείς ορίζουν το συγκεκριμένο γράφημα ως $\mathcal{G} = (V, E)$, με $V = M \cup C \cup A$, όπου:

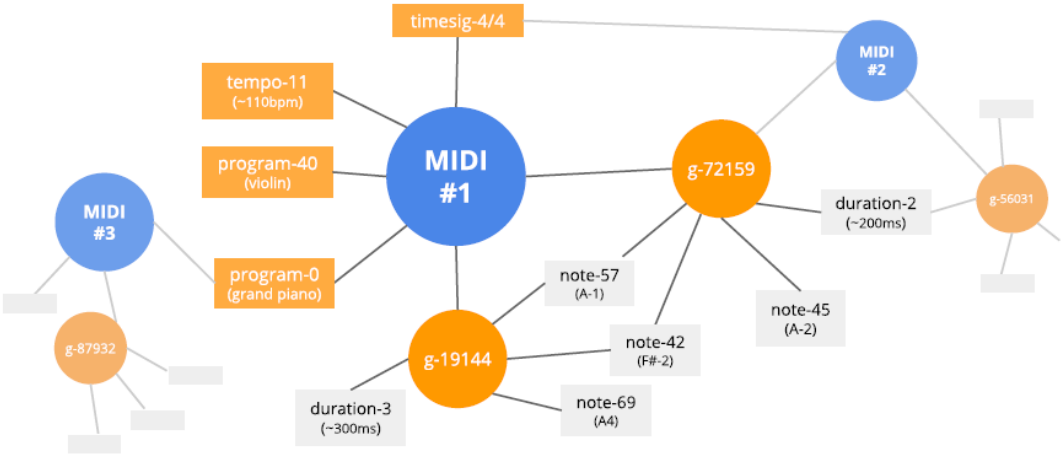
- M είναι το σύνολο αρχείων MIDI,
- C ορίζεται ως το περιεχόμενο MIDI πρώτου επιπέδου, ιδίως ο ρυθμός, τα προγράμματα, η χρονική υπογραφή και το σύνολο των ομάδων νοτών $N \subset C$,
- A είναι το σύνολο των χαρακτηριστικών των ομάδων κόμβων, δηλαδή η διάρκεια, η ταχύτητα και ο τόνος.

και $E = E_M \cup E_N$, όπου:

- η ακμή $(m, c) \in E_M, m \in M, c \in C$ ορίζεται ως `MIDI has content`, ενώ περιέχει επίσης την ακμή $(m, n) \in E_M, m \in M, n \in N$, όπως `MIDI has note group`,
- η ακμή $(n, a) \in E_N, n \in N, a \in A$ γνωστή ως `note group has attribute`.

Όπως βλέπουμε, έχουν οριστεί οι κατηγορίες και η παραπάνω υλοποίηση θυμίζει δομή ετερογενούς γράφου. Ωστόσο, θα προσπαθήσουμε να φτιάξουμε έναν αυστηρό ορισμό για τον ετερογενή γράφο και να μεταφράσουμε αυτή τη δομή στην Python για την πραγματοποίηση των πειραμάτων μας.

[2]Εξίσωση 6.2, καθώς και όλοι οι τυπικοί αριθμοί MIDI μπορούν να βρεθούν στο: https://www.inspiredacoustics.com/en/MIDI_note_numbers_and_center_frequencies.



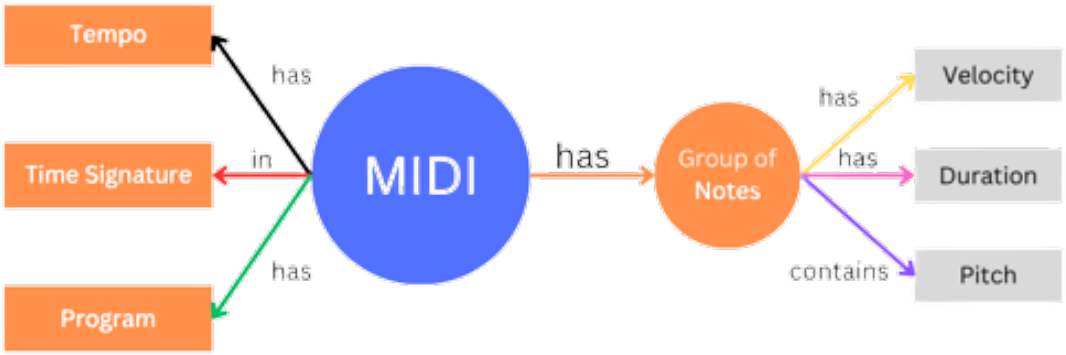
Σχήμα 6.2: Ένα παράδειγμα ενός απλού γράφου MIDI, που περιέχει 3 αρχεία MIDI, με τις σχετικές συνδέσεις τους. Πηγή: [41][Σχήμα 2, σελ. 363]

6.1.2 Ομογενής σε Ετερογενής

Ορίζουμε την ετερογενή εκδοχή του παραπάνω γράφου, ως έναν ετερογενή γράφο $\mathcal{H} = (V, E)$, με σχήμα γράφου $\mathcal{S}_{\mathcal{H}} = (\mathcal{R}, \mathcal{A})$, όπου:

- Το σύνολο των κατηγοριών κόμβων $\mathcal{R} = \{$`MIDI`, `duration`, `note_group`, `pitch`, `program`, `tempo`, `time_sig`, `velocity`$\}$
- Το σύνολο των κατηγοριών ακμών $\mathcal{A} = \{$`MIDI__has__tempo`, `MIDI__in__time_sig`, `MIDI__has__program`, `MIDI__has__note_group`, `note_group__has__velocity`, `note_group__has__duration`, `note_group__contains__pitch`$\}$

Παρατηρήστε ότι οι ακμές που ορίζονται παραπάνω, επιβάλλουν κατεύθυνση στο γράφημά μας. Αυτό μπορεί να φανεί στην ενημερωμένη μορφή του Σχήματος 6.1, το Σχήμα 6.3.



Σχήμα 6.3: Το σχήμα του ετερογενούς γράφου MIDI. Μπορούμε επίσης να δούμε την κατεύθυνση που δημιουργείται με τον ορισμό διαφορετικών τύπων ακμών. Τροποποιημένη εκδοχή του Σχήματος 6.1.

Περισσότερα για τη μετατροπή από το ομοιογενές στο ετερογενές μοντέλο, θα καλυφθούν στο επόμενο κεφάλαιο.

6.2 Αρχιτεκτονική Νευρωνικού Δικτύου

Όπως αναφερθήκαμε εκτενώς σε προηγούμενα κεφάλαια, η επιλογή της αρχιτεκτονικής του μοντέλου, καθώς και της συνάρτησης κόστους και του αλγορίθμου βελτιστοποίησης, εξαρτάται σε μεγάλο βαθμό από την εργασία και τα δεδομένα που έχουμε στη διάθεσή μας. Μετά από πολλούς πειραματισμούς και προσαρμογή των υπερπαραμέτρων, καταλήξαμε στην ακόλουθη διάταξη για το μοντέλο μας.

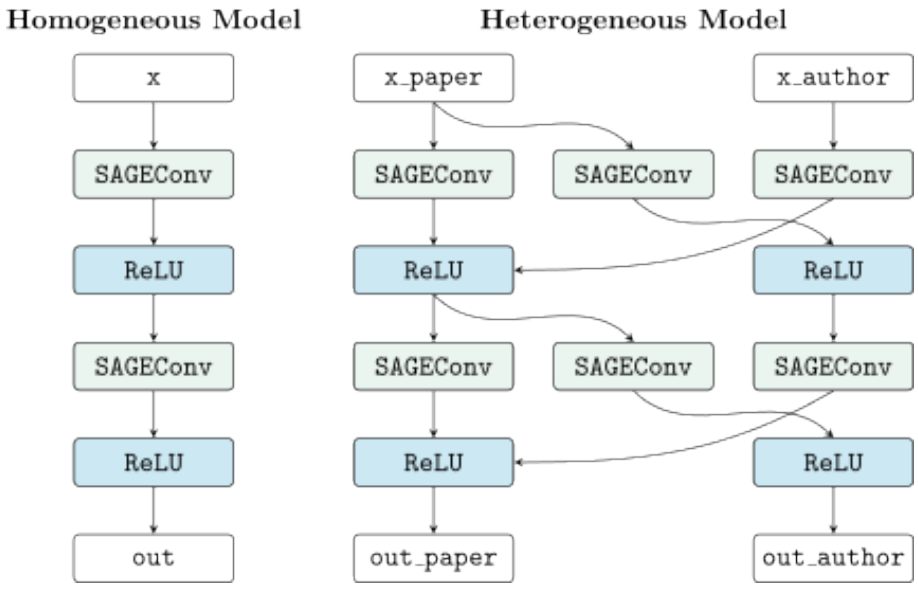
Θα χρησιμοποιήσουμε το συνελικτικό δίκτυο GraphSAGE, όπως παρουσιάζεται στο [24]. Το μοντέλο αποτελείται από 2 επίπεδα GraphSAGE, ακολουθούμενα από μια συνάρτηση ReLU. Υπάρχουν 64 κρυφοί νευρώνες σε κάθε στρώμα και το στρώμα εξόδου έχει έναν νευρώνα ανά κλάση. Θα γίνουν συνολικά τρία πειράματα, με δύο διαφορετικά σύνολα δεδομένων, τα οποία θα παρουσιαστούν αναλυτικά στο επόμενο κεφάλαιο. Στο πρώτο πείραμα, το οποίο είναι ταξινόμηση 5 κλάσεων, θα χρησιμοποιήσουμε 5 νευρώνες εξόδου, στο δεύτερο (10 κλάσεων) 10 νευρώνες και στο τρίτο (15 κλάσεων) 15 νευρώνες.

Όπως αναφέραμε ήδη στο κεφάλαιο 4, η επιλογή του GraphSAGE έγινε επειδή η υλοποίηση μας τείνει να παράγει μεγάλους γράφους, με χιλιάδες κόμβους και εκατομμύρια ακμές. Επιπλέον, τείνουν να έχουν διαφορετικούς βαθμούς κόμβων με βάση τις κατηγορίες τους. Για παράδειγμα, κάθε κόμβος της κατηγορίας `note_group` συνδέεται με πολύ μεγάλο αριθμό άλλων κόμβων, σε σύγκριση με τους κόμβους της κατηγορίας `tempo`. Λόγω των τεχνικών δειγματοληψίας και συνάθροισης, το GraphSAGE μπορεί να ξεπεράσει εύκολα και τα δύο προβλήματα, καθώς μπορεί να μάθει αποτελεσματικά αναπαραστάσεις κόμβων συγκεντρώνοντας πληροφορίες από τοπικές γειτονιές, χωρίς να χρειάζεται να ελέγξει ολόκληρο το γράφημα.

Ο αλγόριθμος βελτιστοποίησης της επιλογής μας είναι ο NAdam[16], καθώς βοηθάει το μοντέλο να αποφύγει εσφαλμένα ελάχιστα της συνάρτησης κόστους, με ρυθμούς μάθησης $\gamma = 0.02$ για τα πειράματα στο SLAC, και $\gamma = 0.01$ για το GiantMIDI-Piano. Η συνάρτηση κόστους που επιλέξαμε είναι η Cross-Entropy Loss, καθώς παρέχει ένα ομαλό και κυρτό τοπίο βελτιστοποίησης, διευκολύνοντας τη βελτιστοποίηση με αλγορίθμους κλίσης, όπως ο NAdam. Προστέθηκε επιπλέον ένας όρος L_2 Regularization, με συντελεστή 5×10^{-3}. Τέλος, κάναμε χρήση της αρχικοποίησης παραμέτρων He (He initialization)[27].

Δυστυχώς, τα περισσότερα μοντέλα έχουν κατασκευαστεί για ομογενείς γράφους και πιθανόν δεν θα είναι συμβατά με ετερογενείς γράφους, λόγω πιθανών διαφορών στη διάσταση των διανυσμάτων χαρακτηριστικών για διαφορετικές κατηγορίες. Μπορούμε εύκολα όμως να επεκτείνουμε το μοντέλο μας σε ετερογενή δεδομένα, χρησιμοποιώντας τη συνάρτηση `to_hetero`[4], της βιβλιοθήκης PyTorch Geometric (PyG)[18]. Με βάση τα docs τους, η `to_hetero` λειτουργεί με την αντιγραφή κάθε στρώματος μετάδοσης

[3] https://pytorch-geometric.readthedocs.io/en/latest/_images/to_hetero.svg
[4] https://pytorch-geometric.readthedocs.io/en/latest/modules/nn.html



Σχήμα 6.4: Μετατροπή ομογενούς μοντέλου σε ετερογενές. Μπορούμε να δούμε ότι το στρώμα μεταβίβασης μηνυμάτων αντιγράφεται για κάθε τύπο ακμής για τη σύνδεση διαφορετικών τύπων κόμβων μεταξύ τους. Πηγή: [3]

μηνυμάτων, μία φορά για κάθε τύπο ακμής που υπάρχει στο γράφημά μας. Αυτό επιτρέπει σε κόμβους διαφορετικών τύπων να ανταλλάσσουν μηνύματα μεταξύ τους. Στη συνέχεια, τα μηνύματα αυτά συγκεντρώνονται και υπολογίζεται η αναπαράσταση του κόμβου. Αυτό μπορεί να γίνει καλύτερα κατανοητό μέσω του σχήματος 6.4, το οποίο προέρχεται από την ίδια σελίδα και παρουσιάζει τη μετατροπή του μοντέλου για τον ετερογενή γράφο του Ακαδημαϊκού Δικτύου, που παρουσιάσαμε στο Κεφάλαιο 2.

Μετά τη μετατροπή λοιπόν, η εξίσωση (4.10) παίρνει την τελική μορφή[5]:

$$\mathbf{h}_u^{(k)} = \sigma\left(\sum_{r\in\mathcal{R}} \mathbf{W}_1^{(r)}\,\mathbf{h}_u^{(k-1)} + \mathbf{W}_2^{(r)} \cdot mean(\{\mathbf{h}_v^{(k-1)},\ \forall v \in \mathcal{N}^{(r)}(u)\})\right) \tag{6.3}$$

όπου $\mathcal{R}$ είναι το σύνολο κατηγοριών ακμών του ετερογενή γράφου $\mathcal{H}$, και $\mathcal{N}^{(r)}(u)$ είναι η γειτονιά του κόμβου u, μέσω των ακμών με κατηγορία $r \in \mathcal{R}$.

Μπορούμε τώρα να προχωρήσουμε στη παρουσίαση των πειραμάτων, όπου εφαρμόζουμε όλες τις διαφορετικές τεχνικές που συζητήθηκαν μέχρι αυτό το σημείο.

[5] https://github.com/pyg-team/pytorch_geometric/discussions/3819

Κεφάλαιο 7

Πειράματα

7.1 Σύνολα δεδομένων & Προετοιμασίες

Τα πειράματα που θα πραγματοποιήσουμε περιλαμβάνουν 2 σύνολα δεδομένων, τα οποία θα παρουσιαστούν παρακάτω: το σύνολο **SLAC** και το σύνολο **GiantMIDI-Piano**.

7.1.1 SLAC

Το πρώτο σύνολο δεδομένων που θα χρησιμοποιήσουμε είναι το σύνολο δεδομένων SLAC[48]. Το SLAC αποτελείται από 250 διαφορετικά τραγούδια, σε μορφή MIDI, τα οποία χωρίζονται εξίσου σε 5 μουσικά είδη, των 50 τραγουδιών το καθένα: **Blues**, **Classical**, **Jazz**, **Rap** και **Rock**. Στη συνέχεια, κάθε είδος χωρίζεται εξίσου σε δύο υποκατηγορίες, αποτελούμενες από 25 τραγούδια η καθεμία:

- **Modern Blues** και **Traditional Blues** σαν υποσύνολα της Blues,
- **Baroque** και **Romantic** σαν υποσύνολα της Classical,
- **Bop** και **Swing** σαν υποσύνολα της Jazz,
- **Hardcore Rap** και **Pop Rap** σαν υποσύνολα της Rap, και
- **Alternative Rock** και **Metal** σαν υποσύνολα της Rock.

Αυτό μας δίνει τη δυνατότητα να εκτελέσουμε 2 διαφορετικές εργασίες ταξινόμησης: μία 5 τάξεων και μία 10 τάξεων. Ενώ το σύνολο δεδομένων έχει περιορισμένο μέγεθος, λόγω του τρόπου με τον οποίο λειτουργεί η δημιουργία του γράφου, παράγει έναν σημαντικά μεγάλο γράφο (93553 κόμβοι και 786635 ακμές). Επιπλέον, μπορούμε να έχουμε μια αρχική εικόνα για τα αποτελέσματά μας συγκρίνοντας τα με τα αποτελέσματα από το [41].

7.1.2 GiantMIDI-Piano

Το δεύτερο σύνολο δεδομένων είναι ένα υποσύνολο του συνόλου δεδομένων GiantMIDI-Piano[40]. Αυτό το πλήρες σύνολο αποτελείται από 10.855 αρχεία MIDI, τα οποία ανήκουν σε 2.786 συνθέτες. Επιπλέον, υπάρχει ένα επιμελημένο υποσύνολο 7.236 MIDI και 1.787 συνθετών, το οποίο περιορίζει τα επώνυμα των συνθετών. Θα χρησιμοποιήσουμε ένα υποσύνολο αυτού του επιμελημένου υποσυνόλου. Πιο συγκεκριμένα, πρώτα φιλτράρουμε για συνθέτες που έχουν περισσότερα από 50 MIDIs στο όνομά τους. Αυτοί οι συνθέτες παρουσιάζονται στον πίνακα 7.1, μαζί με τον αριθμό των αρχείων MIDI στο υποσύνολο. Από αυτά τα 1.238 αρχεία, επιλέγουμε τα 50 πρώτα (με αλφαβητική σειρά) του κάθε συνθέτη για να σχηματίσουμε το τελικό υποσύνολο των 750 αρχείων MIDI από 15 συνθέτες. Το έργο μας θα είναι να εκτελέσουμε μια ταξινόμηση 15 κατηγοριών για τον συνθέτη κάθε κομματιού.

Συνθέτης	MIDI αρχεία
Bach	129
Beethoven	76
Carbajo	77
Chopin	96
Czerny	58
Handel	57
Haydn	54
Liszt	141
Mozart	72
Rebikov	54
Satie	52
Scarlatti	140
Schubert	96
Scriabin	67
Simpson	69

Πίνακας 7.1: Συνθέτες με $\geq$ 50 αρχεία MIDI στο σύνολο δεδομένων GiantMIDI-Piano.

7.1.3 Προετοιμασίες

Και τα δύο σύνολα δεδομένων προετοιμάζονται με την ίδια διαδικασία, με μικρές ίσως διαφορές στις συναρτήσεις που χρησιμοποιούνται. Ο κώδικας που χρησιμοποιήθηκε για την προ-επεξεργασία, καθώς και τα πειράματα, βρίσκονται στο ακόλουθο αποθετήριο GitHub: https://github.com/KottonP/midi2vec-master.

Πρώτον, χρησιμοποιούμε τον αλγόριθμο `midi2edgelist` για να δημιουργήσουμε τα αρχεία `edgelist` για το γράφημά μας. Στη συνέχεια, φορτώνουμε τα edgelists σε έναν γράφο NetworkX[22], εξάγουμε τους κόμβους και τις ακμές `DataFrames`[69] και ονομάζουμε κάθε μεμονωμένο κόμβο και ακμή με βάση τον τύπο τους. Τα ονόματα των κόμβων κωδικοποιούνται από έναν απλό κωδικοποιητή ακέραιων αριθμών, ώστε να μπορούν να χρησιμοποιηθούν από το νευρωνικό δίκτυο. Στη συνέχεια, μετατρέπουμε τις μεμονωμένες ακμές σε δείκτες ακμών (αντί να ορίζονται από το όνομα της πηγής και του στόχου, αναπαρίστανται ως δείκτης πηγής και στόχου) και επισυνάπτουμε σε κάθε κόμβο της κατηγορίας MIDI, την ετικέτα του (το είδος μουσικής που ανήκει ή τον συνθέτη). Στο τέλος, δημιουργούμε τον ετερογενή γράφο, ως ένα αντικείμενο `HeteroData` της βιβλιοθήκης Pytorch Geometric. Μετά τη δημιουργία του ετερογενή γράφου, προσθέτουμε αντίστροφες ακμές για να επιτρέψουμε την αμφίδρομη μετάδοση μηνυμάτων μέσω του δικτύου μας.

Η εκπαίδευση έγινε με χρήση μιας συνάρτησης KFold Cross-Validation που υλοποιήσαμε, με έμπνευση της υλοποίηση που παρουσιάζεται εδώ [1]. Η συνάρτηση αυτή κάνει χρήση της συνάρτησης `KFold`, της βιβλιοθήκης `scikit-learn`[55]. Δυστυχώς δεν μπορέσαμε να χρησιμοποιήσουμε Batch Loaders κατά την εκπαίδευση, λόγω τεχνικών δυσκολιών.

[1] https://github.com/pyg-team/pytorch_geometric/blob/decec47e3c9cd67b8ddbd58a15edd00de0c530ef/benchmark/kernel/train_eval.py#L82-L97

Στη συνέχεια, δημιουργούμε το μοντέλο μας, χρησιμοποιώντας τη συνάρτηση `GraphSAGE` της PyG, η οποία δημιουργεί την αρχιτεκτονική που αναλύσαμε στο Κεφάλαιο 6. Για την δοκιμή του δικτύου στο σύνολο δοκιμής, κατασκευάσαμε μια συνάρτηση ψηφοφορίας μοντέλων, για να προσεγγίσουμε τα αποτελέσματα της συνάρτησης `cross_val_predict` της `scikit-learn`, που χρησιμοποιείται στο [41]. Αυτή η συνάρτηση χρησιμοποιεί τα 10 μοντέλα από το cross-validation και, για κάθε περίπτωση στο σύνολο δοκιμής, επιλέγει το αποτέλεσμα με τις περισσότερες ψήφους, μετρώντας τις εξόδους των μοντέλων. Με αυτά τα αποτελέσματα, δημιουργήσαμε και τους πίνακες συγχύσεως, πάνω σε νέο σύνολο δοκιμής, που δυστυχώς παρουσιάζει κάποια επικάλυψη με τα σύνολα εκπαίδευσης (data leakage). Ωστόσο, ελαχιστοποιούμε τη σοβαρότητα αυτής της επικάλυψης μέσω της ψηφοφορίας.

7.2 Πειράματα & Αποτελέσματα

7.2.1 Πειράματα στο σύνολο SLAC

Τα πρώτα πειράματα που θα παρουσιάσουμε, είναι με το σύνολο δεδομένων SLAC. Επιλέξαμε να χρησιμοποιήσουμε αυτό το σύνολο για να έχουμε μια βασική ιδέα για την απόδοση του μοντέλου μας, συγκρίνοντας τα αποτελέσματά μας με το [41] ενώ παράλληλα είναι μικρό, με σαφή διάκριση των κλάσεων. Δυστυχώς, το μέγεθος του συνόλου δρα και ως αρνητικό, διότι μας αναγκάζει να χρησιμοποιήσουμε ένα μικρό μέρος του συνόλου δεδομένων για δοκιμές και εξαγωγή συμπερασμάτων.

Εκπαιδεύσαμε τα μοντέλα μας με 10-Fold Cross-Validation, για 800 και 1600 εποχές, αντίστοιχα.

Στον πίνακα 7.2, βλέπουμε τα αποτελέσματά μας σε σύγκριση με την μέθοδο MIDI2vec, που ουσιαστικά παράγει εμφυτεύσεις των γράφων MIDI (με χρήση του node2vec), και έπειτα εφαρμόζει ένα απλό νευρωνικό δίκτυο, 2 στρωμάτων.

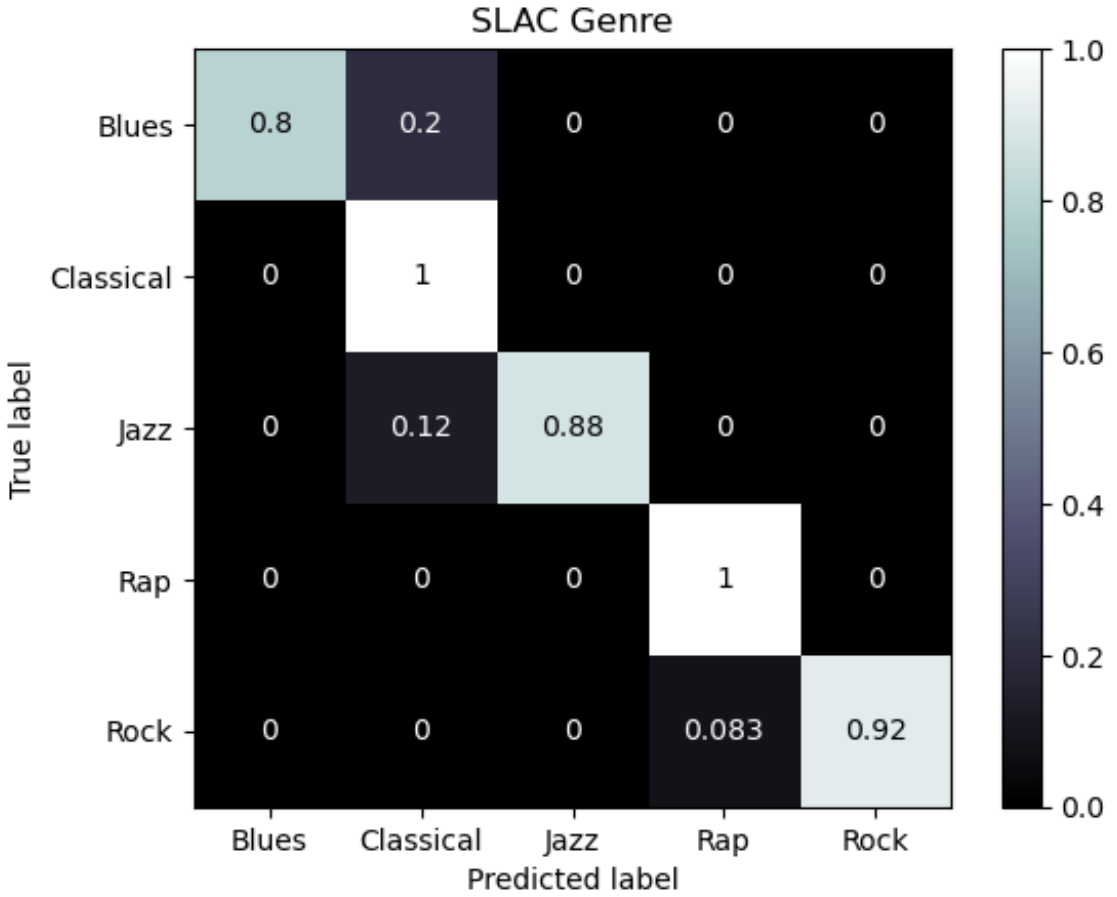
Μοντέλο	5 κλάσεις	10 κλάσεις
MIDI2vec + NN[41]	86.4%(5.4%)	67.2% (7.8%)
Heterogeneous GraphSage	89.6%(8.5%)	74.8% (13.3%)

Πίνακας 7.2: Μέση ακρίβεια μαζί με την τυπική απόκλιση στο 10-Fold Cross-Validation, για το μοντέλο MIDI2vec και το μοντέλο μας στο σύνολο δεδομένων SLAC.

Παρατηρούμε πώς και τα δύο μοντέλα παρουσιάζουν μεγάλη τυπική απόκλιση, λογικά εξαιτίας του συνόλου δεδομένων. Αν και έχουμε ελάχιστα πιο υψηλή ακρίβεια, δεν μπορούμε να χαρακτηρίσουμε το μοντέλο μας πιο αποτελεσματικό, λόγω της διακύμανσης του, που είναι και αυτή λίγο πιο ψιλά.

Όπως αναφέρθηκε, οι πίνακες σύγχυσης που θα παρουσιαστούν, υπολογίστηκαν πάνω σε ένα νέο σύνολο δοκιμής, με μέγεθος $0.2 \times 250 = 50$ αρχεία MIDI και χρήση της συνάρτησης ψηφοφορίας. Αν και παρουσιάζουν καλά αποτελέσματα, δεν μπορούν να

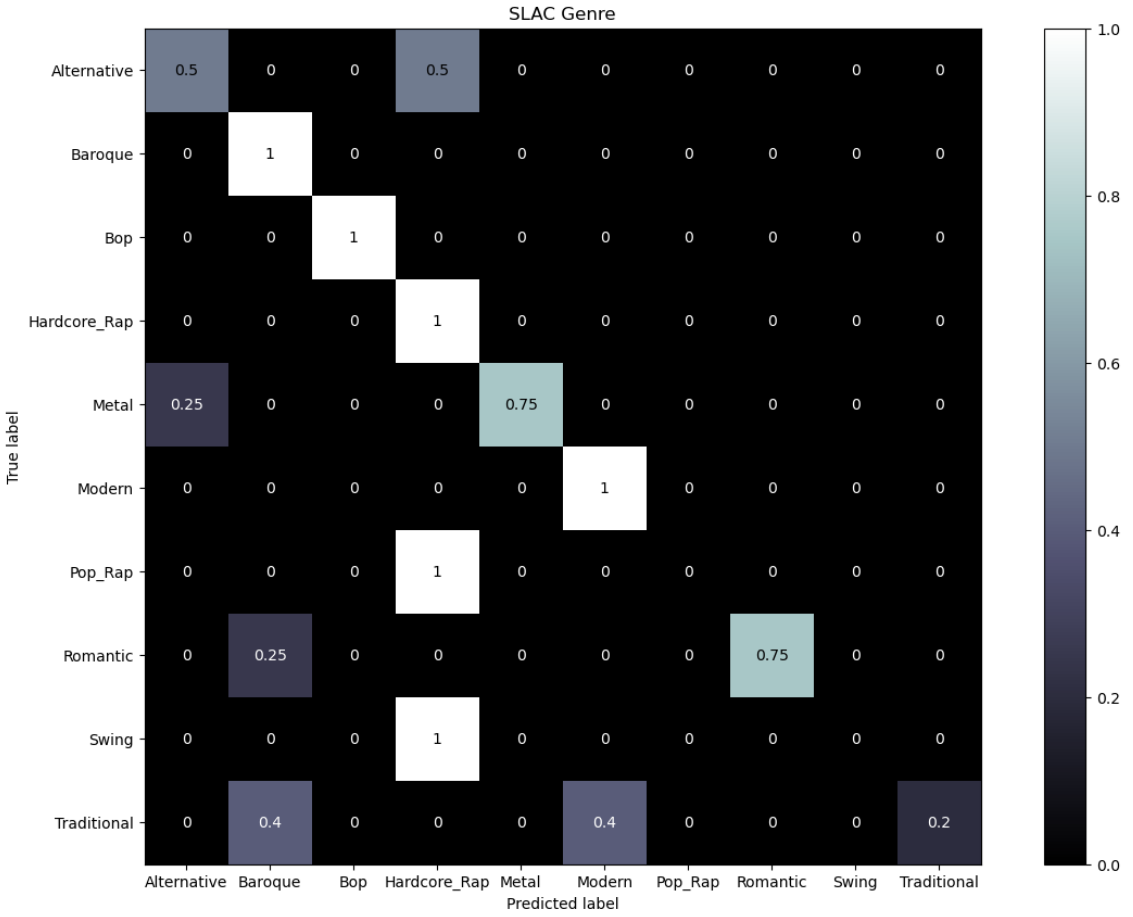
θεωρηθούν πλήρως ενδεικτικά λόγω του μεγέθους και της προαναφερόμενης διακύμανσης του μοντέλου.



Σχήμα 7.1: Ο πίνακας σύγχυσης της ταξινόμησης 5 τάξεων του SLAC, στο σύνολο δοκιμών.

Στο Σχήμα 7.1 βλέπουμε πως το μοντέλο μας έχει καταφέρει να ταξινομήσει σχεδόν τέλεια τα παραδείγματα, με μικρές αποκλίσεις. Στην περίπτωση της σύγχυσης κλασσικής μουσικής με τζαζ, μπορούμε να το δικαιολογήσουμε στις γρήγορες αλλαγές και το γρήγορο tempo των κομματιών, καθώς και στη χρήση παρόμοιων οργάνων, όπως πνευστά, κρουστά και πολύ πιάνο. Στην περίπτωση της λάθος κατηγοριοποίησης blues μουσικής ως κλασσική, ο βασικός λόγος μπορεί είναι ξανά η χρήση κοινών οργάνων και στις δυο κατηγορίες, όπως τα πνευστά και το πιάνο. Επιπλέον, και τα δύο είδη είναι γνωστό πως χαρακτηρίζονται από μελαγχολικά τραγούδια, πληροφορία που μπορεί να έχει συλλάβει ο αλγόριθμός μας. Τέλος, εξίσου η Rock και η Rap, χαρακτηρίζονται από δυναμικά τύμπανα και ρυθμούς, παρόμοια ενορχήστρωση (π.χ. ηλεκτρικές κιθάρες, μπάσο) και παρόμοιες λυρικές μελωδίες, καθώς πολλοί Rock μουσικοί στο παρελθόν έχουν χρησιμοποιήσει Rap φωνητικά, και αντίστροφα.

Τα αποτελέσματα του Σχήματος 7.2 δεν είναι τόσο θετικά, και παρουσιάζουν τις αδυναμίες του μοντέλου μας. Η σύγχυση σε υποκατηγορίες της ίδιας κατηγορίας, όπως οι Pop με Hardcore Rap, Modern με Traditional Blues και Alternative Rock με Metal, είναι λογική, σε ένα μικρό σύνολο. Επιπλέον μπορεί να δικαιολογηθεί η σύγχυση κλασσικής Baroque με Traditional Blues, για τους ίδιους λόγους που αναπτύξαμε παραπάνω



Σχήμα 7.2: Ο πίνακας σύγχυσης για την εργασία ταξινόμησης 10 κατηγοριών του SLAC στο σύνολο δοκιμών.

(παρόμοια στοιχεία σε κλασσική και Blues). Παρουσιάζεται όμως ένα πρόβλημα όταν προσπαθούμε να δικαιολογήσουμε την ταξινόμηση της Swing ως Hardcore Rap. Εκτός απ΄ το ότι και τα δύο είδη έχουν γενικά γρήγορο tempo, με upbeat ρυθμούς, δεν υπάρχουν πολλά άλλα κοινά στοιχεία. Επιπλέον, παρατηρούμε πως το μοντέλο έχει μάθει «πολύ καλά» την κατηγορία της Hardcore Rap, και ταξινομεί εσφαλμένα άλλες κατηγορίες με απόλυτη σιγουριά.

Γενικά, μπορούμε να παρατηρήσουμε πως προβλέπει με σιγουριά τα περισσότερα από τα αποτελέσματα, είτε σωστά, είτε λανθασμένα. Αυτό φαίνεται στο ότι συνήθως οι προβλέψεις έχουν 2 κατηγορίες, με τη μια να υπερισχύει σε ποσοστό, ή απλώνονται το πολύ σε 3 κατηγορίες, με παρόμοια ποσοστά. Ένας πιθανός λόγος είναι η υπερπροσαρμογή σε συγκεκριμένες κλάσεις, που μπορεί να προέρχεται εν μέρη από το μέγεθος του συνόλου δεδομένων και την επικάλυψη που παρατηρείται στα είδη της σύγχρονης μουσικής.

7.2.2 Πειράματα στο GiantMIDI-Piano

Για το τρίτο πείραμα, χρησιμοποιήσαμε ένα υποσύνολο 750 αρχείων MIDI από το σύνολο δεδομένων GiantMIDI-Piano. Για αυτό το πείραμα, εκπαιδεύσαμε το μοντέλο μας με 5-Fold Cross-Validation, για 3200 εποχές. Για το συγκεκριμένο πείραμα προσθέσαμε επιπλέον 2 στρώματα dropout, με πιθανότητες 0.3 το καθένα. Το αποτέλεσμα παρουσιάζεται στον Πίνακα 7.3.

Μοντέλο	15 κλάσεις
Heterogeneous GraphSage	65.3%(3.9%)

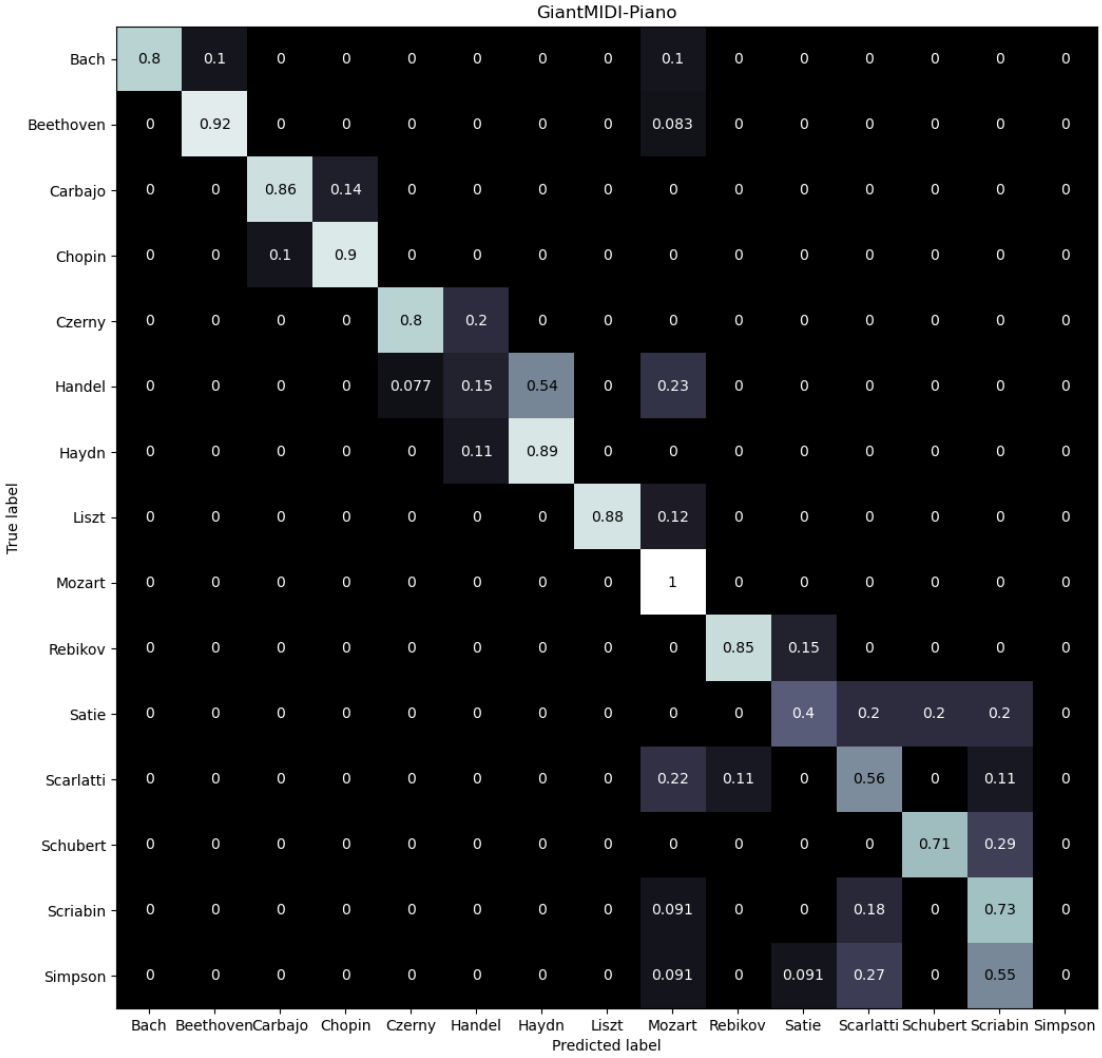
Πίνακας 7.3: Μέση ακρίβεια μαζί με την τυπική απόκλιση στο 5-Fold Cross-Validation, για το μοντέλο μας στο υποσύνολο του GiantMIDI-Piano.

Παρατηρούμε μειωμένη ακρίβεια, λόγω των περισσότερων κλάσεων, αλλά και μειωμένη διακύμανση. Αυτό μας αποκαλύπτει πως όντως η υψηλή διακύμανση στα προηγούμενα πειράματα οφείλεται λογικά στο μικρό μέγεθος του SLAC.

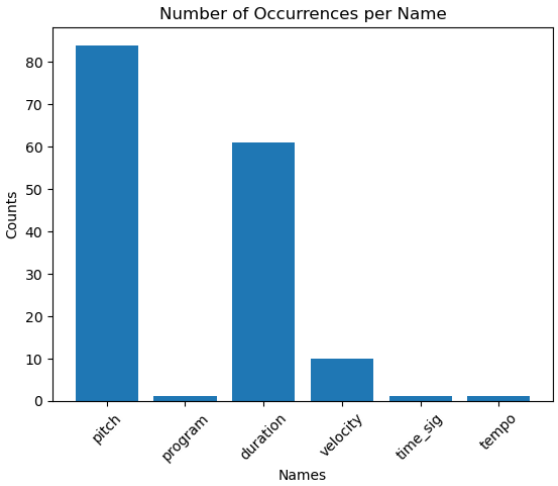
Στο Σχήμα 7.3 παραθέτουμε τον πίνακα σύγχυσης του πειράματος για την ταξινόμηση των αρχείων σε 15 συνθέτες.

Αν και περισσότερες οι κλάσεις της ταξινόμησης, λόγω του μεγαλύτερου μεγέθους των παραδειγμάτων εκπαίδευσης, βλέπουμε πως το μοντέλο παρουσιάζει μεγαλύτερη επιτυχία. Βέβαια, είναι εμφανής η υπερπροσαρμογή του σε συγκεκριμένες κατηγορίες, όπως ο Mozart, που είναι και ο μόνος που έχει πετύχει 100% και παράλληλα έχει ταξινομήσει εσφαλμένα δουλειές άλλων 7 συνθέτων, σαν δικές του. Επιπλέον, βλέπουμε πως συμβαίνει και το αντίθετο, όπως στην περίπτωση του Simpson, που δεν βρήκε καμία σύνθεση του και έχει ταξινομήσει όλες τις δουλειές του σε άλλους συνθέτες. Γενικά, λόγω του είδους της μουσικής και των διαθέσιμων οργάνων, είναι κατανοητό να υπάρχει μεγαλύτερη σύγχυση μεταξύ των συνθετών, αφού όλοι έδρασαν σε σχετικά κοντινές χρονολογίες, με σημαντικές επιρροές ανάμεσά τους.

Επιπλέον, κοιτώντας το Σχήμα 7.4, βλέπουμε πως κατηγορίες όπως το μέτρο του τραγουδιού, το tempo και το πρόγραμμα οργάνου απαρτίζονται εξ' ολοκλήρου από έναν κόμβο (π.χ. `ts-4/4`), πράγμα που δεν προσφέρει έξτρα πληροφορία στο μοντέλο μας. Συνοψίζοντας, τα αποτελέσματα φαίνονται πιο προσεγμένα σε σχέση με το προηγούμενο σύνολο δεδομένων, έχοντας βέβαια ακόμα κάποια άσχημα χαρακτηριστικά, όπως η σιγουριά στις απαντήσεις.



Σχήμα 7.3: Ο πίνακας σύγχυσης για την ταξινόμηση συνθετών GiantMIDI-Piano, στο σύνολο δοκιμών.



Σχήμα 7.4: Αριθμός κόμβων ανά κατηγορία για το σύνολο GiantMIDI-Piano, με εξαίρεση των κατηγοριών `note_group` (285642 στο πλήθος) και `MIDI` (750).

Κεφάλαιο 8

Συμπεράσματα & Μελλοντική Έρευνα

8.1 Συμπεράσματα

Συνοψίζοντας, η παρούσα διατριβή διερεύνησε την εφαρμογή ετερογενών γράφων και νευρωνικών δικτύων γράφων για την ταξινόμηση MIDI. Μέσω μιας σειράς πειραμάτων, λάβαμε πολλά υποσχόμενα αποτελέσματα που αποδεικνύουν τις δυνατότητες αυτής της προσέγγισης στην καταγραφή των πολύπλοκων σχέσεων και δομών που υπάρχουν στα δεδομένα.

Τα πειράματα που διεξήχθησαν, σε δύο διαφορετικά σύνολα δεδομένων, έδειξαν την αποτελεσματικότητα της αξιοποίησης ετερογενών γράφων για τη μοντελοποίηση των ποικίλων πτυχών των συνθέσεων MIDI, όπως ακολουθίες νοτών, πληροφορίες χρονισμού και μουσικού περιεχομένου. Χρησιμοποιώντας τα νευρωνικά δίκτυα γράφων, μπορέσαμε να εξάγουμε ουσιαστικές αναπαραστάσεις από τα δεδομένα MIDI, επιτρέποντας την ακριβή ταξινόμηση διαφορετικών μουσικών ειδών και ταυτοτήτων συνθετών.

Παρόλο που τα αποτελέσματα που προέκυψαν ήταν αρκετά ικανοποιητικά, είναι σημαντικό να αναγνωριστεί ότι υπάρχουν ακόμη περιθώρια βελτίωσης. Χρειάζονται περαιτέρω έρευνες και δοκιμές για να ενισχυθεί η απόδοση της προτεινόμενης μεθοδολογίας. Για παράδειγμα, η διερεύνηση διαφορετικών αρχιτεκτονικών γράφων, η ενσωμάτωση επιπλέον πληροφοριών ή ο πειραματισμός με εναλλακτικές παραλλαγές νευρωνικών δικτύων, θα μπορούσαν ενδεχομένως να αποφέρουν ακόμη καλύτερη ακρίβεια ταξινόμησης και δυνατότητες γενίκευσης.

Επιπλέον, η επεκτασιμότητα και η αποτελεσματικότητα της προσέγγισης μπορούν να βελτιστοποιηθούν περαιτέρω. Καθώς το μέγεθος και η πολυπλοκότητα των συνόλων δεδομένων MIDI συνεχίζουν να αυξάνονται, είναι κρίσιμο να γίνει χρήση τεχνικών που μπορούν να χειριστούν ετερογενείς γράφους μεγαλύτερης κλίμακας χωρίς να θυσιάζεται η υπολογιστική αποδοτικότητα.

Μέσω των πειραμάτων μας, είδαμε καθαρά πως το μοντέλο είναι ικανό να εξάγει σημαντικές πληροφορίες από τα αρχεία, αλλά δυσκολεύεται στη χρήση αυτών των πληροφοριών για ταξινόμηση. Εστιάζοντας την προσοχή μας στο δεύτερο πείραμα, παρατηρήσαμε πως παρουσιάζει μεγάλη σιγουριά στις αποφάσεις του, ακόμα και αν αυτές είναι λανθασμένες. Αυτό μπορεί να οφείλεται στην υψηλή διακύμανση που προκύπτει λόγω της μορφής του συνόλου δεδομένων, καθώς και της αρχιτεκτονικής του μοντέλου.

Ειδικά κατά τον έλεγχο στο σύνολο ελέγχου, παρατηρήθηκε μεγάλη απόκλιση σε ορισμένες κλάσεις, όπου ταξινομήθηκαν πλήρως εσφαλμένα. Συνήθως αυτά τα σφάλματα είχανε κάποια φυσική σημασία (υποκατηγορίες της ίδιας κατηγορίας, υποκατηγορίες με παρόμοια χαρακτηριστικά), αλλά δεν μπορούμε να είμαστε σίγουροι αν οφείλονται κατεξοχήν σε αυτούς τους λόγους εξαιτίας του μικρού μεγέθους του υποσυνόλου ελέγχου του SLAC. Συνολικά, τα αποτελέσματα παρουσιάζονται συγκρίσιμα με αυτά του [41], χωρίς όμως να μπορούμε να πούμε με σιγουριά πως η μέθοδος μας υπερτερεί στην παρούσα μορφή της.

Τελικά, τα ευρήματα που παρουσιάζονται σε αυτή τη διατριβή αποδεικνύουν τις δυνατότητες των ετερογενών γράφων σε συνδυασμό με τα νευρωνικά δίκτυα γράφων για την ταξινόμηση MIDI. Περαιτέρω έρευνα και ανάπτυξη είναι απαραίτητες για την πλήρη απελευθέρωση της δύναμης αυτής της προσέγγισης και την αντιμετώπιση των

προκλήσεων που βρίσκονται μπροστά μας. Συνεχίζοντας να διευρύνουμε τα όρια αυτού του πεδίου, μπορούμε να ανοίξουμε το δρόμο για καινοτόμες εφαρμογές στην ανάκτηση μουσικών πληροφοριών και να συμβάλουμε στη βαθύτερη κατανόηση της πλούσιας μουσικής γλώσσας που είναι κωδικοποιημένη στις συνθέσεις MIDI.

8.2 Μελλοντική Έρευνα

Κλείνοντας, θα προτείνουμε ορισμένους μελλοντικούς άξονες για περαιτέρω βελτίωση και εξερεύνηση στον τομέα της ταξινόμησης MIDI, καθώς και επεκτάσεις σε άλλους τομείς, όπως τα συστήματα συστάσεων.

Πρώτον, υπάρχει δυνατότητα βελτίωσης των πληροφοριών που συλλέγονται από τα αρχεία MIDI. Επί του παρόντος, η ανάλυση επικεντρώνεται στη μουσική δομή και τα χαρακτηριστικά του MIDI. Η ενσωμάτωση εξωτερικών πηγών δεδομένων, όπως δημόσιες μουσικές βάσεις δεδομένων και οντολογίες, π.χ. Music Ontology[60], μπορεί να παρέχει πολύτιμο περιεχόμενο και να ενισχύσει την κατανόηση των συνθέσεων MIDI.

Επιπλέον, μπορεί να διερευνηθεί η χρήση χρονικών γραφημάτων για την ενσωμάτωση πληροφοριών που σχετίζονται με το χρόνο και υπάρχουν στο αρχείο MIDI. Οι αρχιτεκτονικές χρονικών γράφων μπορούν να συλλάβουν τη δυναμική και τις χρονικές εξαρτήσεις που υπάρχουν στη μουσική, επιτρέποντας μια πιο λεπτομερή ανάλυση των μουσικών μοτίβων και της εξέλιξης με την πάροδο του χρόνου.

Όσον αφορά τα νευρωνικά δίκτυα γράφων, μπορούν να διερευνηθούν εναλλακτικοί τύποι και αρχιτεκτονικές για την περαιτέρω βελτίωση της απόδοσης της ταξινόμησης MIDI. Μπορούν να διερευνηθούν αρχιτεκτονικές καθαρά ετερογενών δικτύων, οι οποίες μοντελοποιούν ρητά την ετερογένεια και τις πλούσιες αλληλεπιδράσεις μεταξύ διαφορετικών τύπων κόμβων και ακμών στον γράφο MIDI. Παραδείγματα τέτοιων αρχιτεκτονικών περιλαμβάνουν τα Heterogeneous Graph Convolutional Networks (HetGCN)[59] και τα Graph Attention Networks (GAT)[32] προσαρμοσμένα για ετερογενείς γράφους.

Επιπλέον, η ενσωμάτωση τεχνικών συστημάτων συστάσεων στο πλαίσιο ταξινόμησης MIDI μπορεί να επεκτείνει τη δυναμική του. Με την εισαγωγή πρόσθετων κατηγοριών κόμβων, όπως χρήστες ή ακροατές, και τη χρήση τεχνικών πρόβλεψης συνδέσμων, το μοντέλο μπορεί να επεκταθεί για τη δημιουργία εξατομικευμένων μουσικών συστάσεων με βάση την ανάλυση των γράφων MIDI.

Ανακεφαλαιώνοντας, η μελλοντική έρευνα έγκειται στην επέκταση του πεδίου συλλογής πληροφοριών, στη διερεύνηση διαφορετικών αρχιτεκτονικών GNN, στην ενσωμάτωση χρονικών πτυχών και στην επέκταση του πλαισίου σε συστήματα συστάσεων. Αυτές οι εξελίξεις θα βελτιώσουν περαιτέρω την κατανόηση και την ανάλυση των συνθέσεων MIDI, ανοίγοντας πόρτες σε νέες δυνατότητες στους τομείς της ανάκτησης μουσικών πληροφοριών και των μουσικών συστάσεων.

Bibliography

[1] James Atwood and Don Towsley. Diffusion-Convolutional Neural Networks, July 2016. arXiv:1511.02136 [cs].

[2] The TensorFlow Authors. TensorFlow Documentation. https://github.com/tensorflow/docs/blob/0c2cf1d56fbd9ce171f96a5121554126a0fef903/site/en/tutorials/images/transfer_learning.ipynb. original-date: 2018-04-12T22:23:12Z, Accessed: 2023-05-18.

[3] Aharon Azulay and Yair Weiss. Why do deep convolutional networks generalize so poorly to small image transformations? *Journal of Machine Learning Research*, 20(184):1–25, 2019.

[4] Sunitha Basodi, Chunyan Ji, Haiping Zhang, and Yi Pan. Gradient amplification: An efficient way to train deep neural networks. *Big Data Mining and Analytics*, 3(3):196–207, September 2020. Conference Name: Big Data Mining and Analytics.

[5] Edward A. Bender and S. Gll Williamson. Basic Concepts in Graph Theory. In *Lists, Decisions and Graphs*. S. Gill Williamson, 2010. Google-Books-ID: vaXv_yhefG8C.

[6] Etienne Bernard. *Introduction to Machine Learning*. Wolfram Media Inc, Champaign, December 2021. Accessed through https://www.wolfram.com/language/introduction-machine-learning/machine-learning-paradigms/, at 19/06/2023.

[7] Christopher M. Bishop. *Pattern recognition and machine learning*, chapter 3, pages 144–145. Information science and statistics. Springer, New York, 2006.

[8] Wadii Boulila, Maha Driss, Mohamed Al-Sarem, Faisal Saeed, and Moez Krichen. Weight Initialization Techniques for Deep Learning Algorithms in Remote Sensing: Recent Trends and Future Perspectives, February 2021. arXiv:2102.07004 [cs].

[9] Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges, May 2021. arXiv:2104.13478 [cs, stat].

[10] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation, September 2014. arXiv:1406.1078 [cs, stat].

[11] Wikipedia Contributors. Feedforward neural network. https://en.wikipedia.org/w/index.php?title=Feedforward_neural_network&oldid=1155597882#cite_note-Zell1994p73-1. Accessed: 2023-06-01, Page Version ID: 1155597882.

[12] Wikipedia Contributors. Softmax function. https://en.wikipedia.org/w/index.php?title=Softmax_function&oldid=1150116337. Accessed: 2023-05-16, Page Version ID: 1150116337.

[13] Wikipedia Contributors. A440 (pitch standard). https://en.wikipedia.org/w/index.php?title=A440_(pitch_standard)&oldid=1130168302, December 2022. Accessed: 2023-06-14, Page Version ID: 1130168302.

[14] Débora C. Corrêa and Francisco Ap. Rodrigues. A survey on symbolic data-based music genre classification. *Expert Systems with Applications*, 60:190–210, October 2016.

[15] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering, February 2017. arXiv:1606.09375 [cs, stat].

[16] T. Dozat. Incorporating Nesterov Momentum into Adam. In *Proceedings of the 4th International Conference on Learning Representations*, February 2016.

[17] John Duchi, Elad Hazan, and Yoram Singer. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *The Journal of Machine Learning Research*, 12(null):2121–2159, July 2011.

[18] Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

[19] M. Gavish, B. Nadler, and R. Coifman. Multiscale Wavelets on Trees, Graphs and High Dimensional Data: Theory and Applications to Semi Supervised Learning. In *International Conference on Machine Learning*, June 2010.

[20] Aditya Grover and Jure Leskovec. node2vec: Scalable Feature Learning for Networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 855–864, San Francisco California USA, August 2016. ACM.

[21] Aurélien Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: concepts, tools, and techniques to build intelligent systems*, chapter 2, pages 211–214. O'Reilly Media, Inc, Beijing [China] ; Sebastopol, CA, second edition edition, 2019.

[22] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring Network Structure, Dynamics, and Function using NetworkX. In Gaël Varoquaux, Travis Vaught, and Jarrod Millman, editors, *Proceedings of the 7th Python in Science Conference*, pages 11 – 15, Pasadena, CA USA, 2008.

[23] William L Hamilton. Graph Representation Learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 14(3):1–159, 2020.

[24] William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive Representation Learning on Large Graphs, September 2018. arXiv:1706.02216 [cs, stat].

[25] David K. Hammond, Pierre Vandergheynst, and Rémi Gribonval. Wavelets on Graphs via Spectral Graph Theory, December 2009. arXiv:0912.3848 [cs, math].

[26] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. Overview of Supervised Learning. In Trevor Hastie, Robert Tibshirani, and Jerome Friedman, editors, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer Series in Statistics, chapter 2, pages 9–41. Springer, New York, NY, 2009.

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification, February 2015. arXiv:1502.01852 [cs].

[28] Knut Hinkelmann. Neural Networks. https://web.archive.org/web/20181006235506/http://didattica.cs.unicam.it/lib/exe/fetch.php?media=didattica:magistrale:kebi:ay_1718:ke-11_neural_networks.pdf, October 2018. University of Applied Sciences Northwestern Switzerland. Accessed: 2023-05-16.

[29] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors, July 2012. arXiv:1207.0580 [cs].

[30] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, November 1997.

[31] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open Graph Benchmark: Datasets for Machine Learning on Graphs, February 2021. arXiv:2005.00687 [cs, stat].

[32] Qi Huang, Junshuai Yu, Jia Wu, and Bin Wang. Heterogeneous Graph Attention Networks for Early Detection of Rumors on Twitter, June 2020. arXiv:2006.05866 [cs].

[33] I. T. Jolliffe. *Principal Component Analysis*. Springer Series in Statistics. Springer-Verlag, New York, 2002.

[34] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima, February 2017. arXiv:1609.04836 [cs, math].

[35] Representing graphs (article)|*Khan Academy*|Accessed May 4, 2023| https://www.khanacademy.org/computing/computer-science/algorithms/graph-representation/a/representing-graphs.

[36] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization, January 2017. arXiv:1412.6980 [cs].

[37] Thomas N. Kipf and Max Welling. Variational Graph Auto-Encoders, November 2016. arXiv:1611.07308 [cs, stat].

[38] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks, February 2017. arXiv:1609.02907 [cs, stat].

[39] Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th international joint conference on Artificial intelligence - Volume 2*, IJCAI'95, pages 1137–1143, San Francisco, CA, USA, August 1995. Morgan Kaufmann Publishers Inc.

[40] Qiuqiang Kong, Bochen Li, Jitong Chen, and Yuxuan Wang. GiantMIDI-Piano: A large-scale MIDI dataset for classical piano music, April 2022. arXiv:2010.07061 [cs, eess].

[41] Pasquale Lisena, Albert Meroño-Peñuela, and Raphaël Troncy. MIDI2vec: Learning MIDI embeddings for reliable prediction of symbolic music metadata. *Semantic Web*, 13(3):357–377, April 2022.

[42] Lu Lu, Yeonjong Shin, Yanhui Su, and George Em Karniadakis. Dying ReLU and Initialization: Theory and Numerical Examples. *Communications in Computational Physics*, 28(5):1671–1706, June 2020. arXiv:1903.06733 [cs, math, stat].

[43] Laurens van der Maaten and Geoffrey Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.

[44] Stein Malerud. Modelling human social behaviour in conflict environments using complex adaptive systems. *Norwegian Defence Research Establishment (FFI)*, January 2008.

[45] Haggai Maron, Heli Ben-Hamu, Nadav Shamir, and Yaron Lipman. Invariant and Equivariant Graph Networks, April 2019. arXiv:1812.09902 [cs, stat].

[46] Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, December 1943.

[47] C. McKay, J. Burgoyne, Jason Hockman, Jordan B. L. Smith, Gabriel Vigliensoni, and Ichiro Fujinaga. Evaluating the Genre Classification Performance of Lyrical Features Relative to Audio, Symbolic and Cultural Features. In *International Society for Music Information Retrieval Conference*, 2010.

[48] Cory McKay. SLAC Dataset, March 2021.

[49] Alessio Micheli. Neural Network for Graphs: A Contextual Constructive Approach. *IEEE Transactions on Neural Networks*, 20(3):498–511, March 2009. Conference Name: IEEE Transactions on Neural Networks.

[50] Robert A Moog. Midi: Musical instrument digital interface. *Journal of the Audio Engineering Society*, 34(5):394–404, 1986.

[51] Coenraad Mouton, Johannes C. Myburgh, and Marelie H. Davel. Stride and Translation Invariance in CNNs. In *Artificial Intelligence Research*, volume 1342, pages 267–281. Springer Cham, 2020. arXiv:2103.10097 [cs].

[52] Michael Mozer. A Focused Backpropagation Algorithm for Temporal Pattern Recognition. *Complex Systems*, 3, January 1995.

[53] DQ Nykamp. Node degree definition | Accesed May 4, 2023 | https://mathinsight.org/definition/node_degree.

[54] Josh Patterson and Adam Gibson. *Deep learning: a practitioner's approach*, chapter 6, page 258–263. O'Reilly, Sebastopol, CA, first edition edition, 2017. OCLC: ocn902657832.

[55] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[56] R. Penrose. A generalized inverse for matrices. *Mathematical Proceedings of the Cambridge Philosophical Society*, 51(3):406–413, July 1955.

[57] Michael Pilhofer and Holly Day. *Music Theory For Dummies*. John Wiley & Sons, February 2015. Google-Books-ID: 7p6LBgAAQBAJ.

[58] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation, April 2017. arXiv:1612.00593 [cs].

[59] Rahul Ragesh, Sundararajan Sellamanickam, Arun Iyer, Ram Bairi, and Vijay Lingam. HeteGCN: Heterogeneous Graph Convolutional Networks for Text Classification, August 2020. arXiv:2008.12842 [cs, stat].

[60] Yves Raimond, S. Abdallah, M. Sandler, and Frederick Giasson. The Music Ontology. In *International Society for Music Information Retrieval Conference*, 2007.

[61] A. J. Robinson and Frank Fallside. The utility driven dynamic error propagation network. Technical Report CUED/F-INFENG/TR.1, Engineering Department, Cambridge University, Cambridge, UK, 1987.

[62] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, October 1986.

[63] Zahraa Al Sahili and Mariette Awad. Spatio-Temporal Graph Neural Networks: A Survey, February 2023. arXiv:2301.10569 [cs].

[64] Percy Scholes. The Oxford Companion to Music. In Alison Latham, editor, *The Oxford Companion to Music*. Oxford University Press, January 2011.

[65] scikit-learn developers. Stochastic Gradient Descent. https://scikit-learn/stable/modules/sgd.html. Accessed: 2023-05-18.

[66] David I. Shuman, Sunil K. Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. The Emerging Field of Signal Processing on Graphs: Extending High-Dimensional Data Analysis to Networks and Other Irregular Domains. *IEEE Signal Processing Magazine*, 30(3):83–98, May 2013. arXiv:1211.0053 [cs].

[67] Leslie N. Smith. Cyclical Learning Rates for Training Neural Networks, April 2017. arXiv:1506.01186 [cs].

[68] Yizhou Sun and Jiawei Han. Mining heterogeneous information networks: a structural analysis approach. *ACM SIGKDD Explorations Newsletter*, 14(2):20–28, April 2013.

[69] The pandas development team. pandas-dev/pandas: Pandas, February 2020.

[70] The MIDI Manufacturers Association. *The Complete MIDI 1.0 Detailed Specification Document (1996)*. The MIDI Manufacturers Association Los Angeles, CA, January 1996. This publication represents the complete documentation of The MIDI Specification and all related Recommended Practices as of 1996. For subsequent corrections and additions visit https://www.midi.org/specifications.

[71] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31, 2012.

[72] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks, February 2018. arXiv:1710.10903 [cs, stat].

[73] Xiao Wang, Deyu Bo, Chuan Shi, Shaohua Fan, Yanfang Ye, and Philip S. Yu. A Survey on Heterogeneous Graph Embedding: Methods, Techniques, Applications and Sources, November 2020. arXiv:2011.14867 [cs].

[74] Paul J. Werbos. Generalization of backpropagation with application to a recurrent gas market model. *Neural Networks*, 1(4):339–356, January 1988.

[75] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A Comprehensive Survey on Graph Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24, January 2021. arXiv:1901.00596 [cs, stat].

[76] Xin Xin, Alexandros Karatzoglou, Ioannis Arapakis, and Joemon M. Jose. Self-Supervised Reinforcement Learning for Recommender Systems, June 2020. arXiv:2006.05779 [cs].

[77] Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J. Kim. Graph Transformer Networks, February 2020. arXiv:1911.06455 [cs, stat].

[78] Wei Zhang, Jun Tanida, Kazuyoshi Itoh, and Yoshiki Ichioka. Shift-invariant pattern recognition neural network and its optical architecture. In *Proceedings of annual conference of the Japan Society of Applied Physics*, pages 2147–2151. Montreal, CA, 1988.